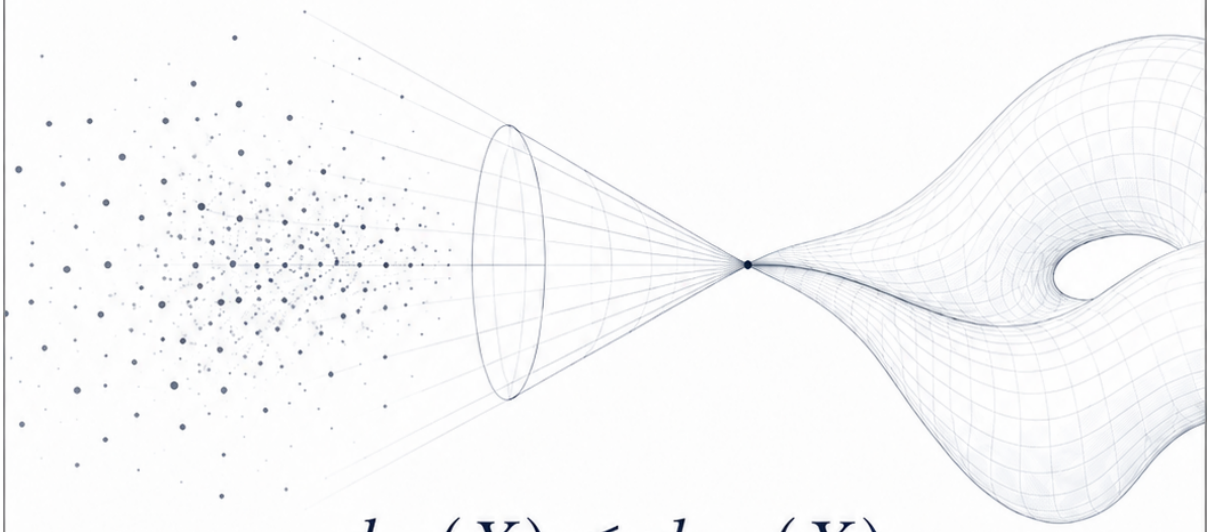
DRCC

Dimensional Reduction via Controlled Combinatorics



$$d_{\mathrm{rec}}(X) \leq d_{\mathrm{frag}}(X)$$

DRCC Criterion

DRCC is proposed as a toolbox for controlled reduction, designed to identify structural principles implicitly present in successful mathematical solution strategies.

DRCC-V2

Reza Hesamiy

Munich, Germany

2026

Notation and Symbols

This condensed manuscript uses the following notation.

Symbol	Meaning
$\mathcal{P}$	Class of admissible problem instances.
$P \in \mathcal{P}$	A finite problem instance.
$X(P)$	Classical candidate space associated with P.
$x \in X(P)$	A candidate solution, state, assignment, route, coloring, or decision option.
$X_{\mathrm{adm}}(P)$	Set of admissible candidates satisfying the constraints of P.
$d_{\mathrm{frag}}(P)$	Fragmentation dimension; $d_{\mathrm{frag}}(P) = \lvert X(P)\rvert$.
C_P	Controlled reduction map $C_P : X(P) \to Z(P)$.
$Z(P)$	Set of reduced structural states associated with P.
$\zeta \in Z(P)$	A reduced state or structural class.
$C_P^{-1}(\zeta)$	Reduction class of all candidates mapped to ζ.
$\Omega_P(\zeta)$	Reconstruction space associated with the reduced state ζ.
$d_{\mathrm{rec}}(P,\zeta)$	Reconstruction dimension; $d_{\mathrm{rec}}(P,\zeta) = \lvert\Omega_P(\zeta)\rvert$.
$\mathcal{R}$	Controlled reconstruction operator.
$N_{\mathrm{Classic}}(P)$	Classical evaluation count.
$N_{\mathrm{Collapse}}(P)$	Collapse evaluation count.
$N_{\mathrm{Reconstruction}}(P,\zeta)$	Reconstruction evaluation count.
$N_{\mathrm{DRCC}}(P,\zeta)$	Total DRCC evaluation count.
$T_{\mathrm{Classic}}(P)$	Classical runtime.
$T_{\mathrm{DRCC}}(P,\zeta)$	DRCC runtime.
$T_{\mathrm{Collapse}}(P)$	Runtime contribution of the collapse phase.
$T_{\mathrm{Reconstruction}}(P,\zeta)$	Runtime contribution of the reconstruction phase.
f	Reference operation rate used to convert counts into runtime.
$G(P,\zeta)$	Runtime gain of DRCC relative to classical enumeration.
$W = (n_w, R_w[s])$	Discrete transition point, consisting of problem size n_w and runtime value $R_w[s]$.
$\Delta_T(n)$	Runtime gap $\Delta_T(n) = T_{\mathrm{Classic}}(n) - T_{\mathrm{DRCC}}(n)$.
$\Delta_N(n)$	Count-level gap $\Delta_N(n) = N_{\mathrm{Classic}}(n) - N_{\mathrm{DRCC}}(n)$.
CDI_δ	Collapse Depth Index in difference form.
$\mathrm{CDI}_{\log}$	Collapse Depth Index in logarithmic form.
$d_{\mathrm{rec}}^{\log}$	Logarithmic reconstruction dimension.

All symbols are used in a finite combinatorial sense unless explicitly stated otherwise. In particular, the reconstruction space $\Omega_P(\zeta)$ is a finite set in this condensed manuscript.

Contents

Abstract

This condensed manuscript presents a runtime-oriented formulation of Dimensional Reduction via Controlled Combinatorics (DRCC–V2).

DRCC is a finite, discrete framework for studying reconstruction problems through controlled structural reduction. The guiding principle is to preserve the information required for admissible reconstruction and to collapse distinctions that are irrelevant to the reconstruction task.

The manuscript defines the classical candidate space $X(P)$, the fragmentation dimension

$$d_{\mathrm{frag}}(P) = |X(P)|,$$

the controlled reduction map

$$C_P : X(P) \to Z(P),$$

the reconstruction space

$$\Omega_P(\zeta) = \{x \in X_{\mathrm{adm}}(P) : C_P(x) = \zeta\},$$

and the reconstruction dimension

$$d_{\mathrm{rec}}(P, \zeta) = |\Omega_P(\zeta)|.$$

The operational stability condition is

$$0 < d_{\mathrm{rec}}(P, \zeta) \leq d_{\mathrm{frag}}(P) < \infty.$$

The runtime model compares classical enumeration with a DRCC decomposition into collapse and reconstruction:

$$T_{\mathrm{DRCC}} = T_{\mathrm{Collapse}} + T_{\mathrm{Reconstruction}}.$$

The runtime gain is defined as

$$G(P, \zeta) = \frac{T_{\mathrm{Classic}}(P)}{T_{\mathrm{DRCC}}(P, \zeta)}.$$

The transition point $W = (n_w, R_w[s])$ marks the discrete size at which the classical and DRCC runtime models coincide.

The manuscript applies this framework to finite case studies including housing selection, travelling-salesman-type routing, constraint satisfaction problems, satisfiability, graph 3-coloring, and the full adder. These examples are used to illustrate how controlled collapse, reconstruction dimension, runtime gain, and transition behavior can be computed or bounded in concrete settings.

The work does not claim a proof of P versus NP. It does not claim a universal speedup for graph coloring, satisfiability, constraint satisfaction, or routing problems. Its contribution is a compact mathematical language and runtime accounting framework for identifying when controlled structural reduction may become advantageous relative to classical enumeration.

1 Core Definitions and Runtime Framework

1.1 Purpose of the Condensed Manuscript

This condensed manuscript focuses on the runtime and structural transition aspects of Dimensional Reduction via Controlled Combinatorics (DRCC).

It is not intended to replace the full DRCC–V2 monograph.

Instead, it addresses reviewer-relevant questions concerning:

1. Precise mathematical definitions of the core DRCC quantities.
2. Explicit runtime models and transition-point formulation.
3. Concrete algorithmic protocol.
4. Compact case studies with runtime evidence.
5. Cross-case comparison tables.
6. Relation to established literature.
7. Explicit limitations and non-claims.

The objective is not to claim universal computational improvement.

The objective is to provide a compact mathematical framework for analyzing when controlled structural reduction and reconstruction may become advantageous relative to classical enumeration.

1.2 Problem Instances and Candidate Spaces

Definition 1.2.1 (Problem Space)**.** Let

$$\mathcal{P} \tag{1.1}$$

denote a class of admissible problem instances.

An element

$$P \in \mathcal{P} \tag{1.2}$$

is called a problem instance.

Examples of problem instances considered in this manuscript include housing selection problems, travelling-salesman-type routing problems, constraint satisfaction problems, Boolean satisfiability instances, graph coloring instances, and structured digital systems.

Definition 1.2.2 (Candidate Space)**.** For a problem instance $P \in \mathcal{P}$, the candidate space is a finite set

$$X(P) = \{x_1, x_2, \ldots, x_N\}. \tag{1.3}$$

Each element

$$x \in X(P) \tag{1.4}$$

is a candidate solution, admissible state, assignment, route, coloring, configuration, or decision option associated with P.

In the classical exhaustive model, solving P requires evaluating all elements of $X(P)$, unless additional structure is used.

Definition 1.2.3 (Fragmentation Dimension)**.** The fragmentation dimension of P is defined by

$$d_{\text{frag}}(P) = |X(P)|. \tag{1.5}$$

In this condensed runtime manuscript, the quantity $d_{\text{frag}}(P)$ is used operationally as the size of the pre-reduction candidate space.

This convention makes the runtime comparison explicit and measurable.

1.3 Controlled Reduction

Definition 1.3.1 (Controlled Reduction Map)**.** Let $P \in \mathcal{P}$ be a problem instance with candidate space $X(P)$.

A controlled reduction map is a mapping

$$C_P : X(P) \to Z(P), \tag{1.6}$$

where $Z(P)$ is the set of reduced structural states.

For a candidate $x \in X(P)$, the value

$$\zeta = C_P(x) \tag{1.7}$$

is called a reduced state or structural class.

The reduction map C_P is not required to be injective.

Indeed, the purpose of DRCC is often to identify candidates that are distinct in the classical candidate space but equivalent with respect to a structural reconstruction task.

Definition 1.3.2 (Reduction Class)**.** For $\zeta \in Z(P)$, the reduction class associated with ζ is

$$C_P^{-1}(\zeta) = \{x \in X(P) : C_P(x) = \zeta\}. \tag{1.8}$$

A reduction class contains all candidates that collapse to the same reduced structural state.

1.4 Reconstruction Spaces

Definition 1.4.1 (Admissible Candidate Space)**.** Let

$$X_{\text{adm}}(P) \subseteq X(P) \tag{1.9}$$

denote the set of candidates that satisfy the admissibility conditions of the problem instance P.

The admissible candidate space may be equal to the full candidate space, or it may be a proper subset determined by constraints.

Definition 1.4.2 (Reconstruction Space)**.** Let $P \in \mathcal{P}$ be a problem instance and let $\zeta \in Z(P)$ be a reduced state.

The reconstruction space associated with ζ is defined by

$$\Omega_P(\zeta) = \{x \in X_{\text{adm}}(P) : C_P(x) = \zeta\}. \tag{1.10}$$

The reconstruction space contains precisely those admissible candidates that are compatible with the reduced state ζ.

Definition 1.4.3 (Reconstruction Dimension)**.** The reconstruction dimension associated with the reduced state ζ is defined by

$$d_{\text{rec}}(P, \zeta) = |\Omega_P(\zeta)|. \tag{1.11}$$

In this condensed runtime manuscript, $d_{\text{rec}}(P, \zeta)$ is used operationally as the size of the admissible reconstruction space after controlled reduction.

1.5 Structural Stability Condition

Definition 1.5.1 (DRCC Structural Stability)**.** A reduced state $\zeta \in Z(P)$ is called structurally stable if

$$0 < d_{\text{rec}}(P, \zeta) \leq d_{\text{frag}}(P) < \infty. \tag{1.12}$$

Equation 1.12 is the operational stability condition used throughout this manuscript.

It excludes three failure modes.

First,

$$d_{\text{rec}}(P, \zeta) = 0$$

means that no admissible reconstruction exists.

Second,

$$d_{\text{rec}}(P, \zeta) > d_{\text{frag}}(P)$$

would contradict the interpretation of reconstruction as a controlled substructure of the original candidate space.

Third,

$$d_{\text{frag}}(P) = \infty$$

would move the analysis outside the finite combinatorial setting of the present manuscript.

Remark 1.5.2 (Operational Nature of the Stability Condition)*.* The condition

$$0 < d_{\text{rec}}(P, \zeta) \leq d_{\text{frag}}(P) < \infty$$

is not a claim that every reduced problem is easy.

It states only that reconstruction remains finite, non-empty, and structurally bounded by the original candidate space.

1.6 Classical Runtime Model

Definition 1.6.1 (Classical Evaluation Count)**.** Let $P \in \mathcal{P}$ be a finite problem instance.
The classical evaluation count is defined by

$$N_{\text{Classic}}(P) = |X(P)| \cdot c_{\text{eval}}(P), \tag{1.13}$$

where $c_{\text{eval}}(P) \geq 1$ denotes the cost of evaluating one candidate in the classical model.

In many examples, the unit-cost model

$$c_{\text{eval}}(P) = 1 \tag{1.14}$$

is used to compare growth behavior.
Under this convention,

$$N_{\text{Classic}}(P) = d_{\text{frag}}(P). \tag{1.15}$$

Definition 1.6.2 (Classical Runtime)**.** Let $f > 0$ denote the reference update rate, measured in operations or state evaluations per second.
The classical runtime is

$$T_{\text{Classic}}(P) = \frac{N_{\text{Classic}}(P)}{f}. \tag{1.16}$$

The reference rate f is introduced only to convert operation counts into wall-clock time.
The asymptotic comparison is governed by the evaluation count $N_{\text{Classic}}(P)$.

1.7 DRCC Runtime Model

The DRCC runtime is decomposed into a reduction phase and a reconstruction phase.

Definition 1.7.1 (Collapse Evaluation Count)**.** The collapse evaluation count is denoted by

$$N_{\text{Collapse}}(P). \tag{1.17}$$

It measures the number of operations required to construct reduced states, structural classes, fragments, or candidate groups.

Definition 1.7.2 (Reconstruction Evaluation Count)**.** For a reduced state ζ, the reconstruction evaluation count is

$$N_{\text{Reconstruction}}(P, \zeta) = d_{\text{rec}}(P, \zeta) \cdot c_{\text{rec}}(P, \zeta), \tag{1.18}$$

where $c_{\text{rec}}(P, \zeta) \geq 1$ denotes the cost of evaluating or constructing one admissible reconstruction.

Definition 1.7.3 (DRCC Evaluation Count)**.** The DRCC evaluation count is

$$N_{\text{DRCC}}(P, \zeta) = N_{\text{Collapse}}(P) + N_{\text{Reconstruction}}(P, \zeta). \tag{1.19}$$

Definition 1.7.4 (DRCC Runtime)**.** The DRCC runtime is

$$T_{\text{DRCC}}(P, \zeta) = \frac{N_{\text{DRCC}}(P, \zeta)}{f}. \tag{1.20}$$

Thus,

$$T_{\mathrm{DRCC}}(P,\zeta) = T_{\mathrm{Collapse}}(P) + T_{\mathrm{Reconstruction}}(P,\zeta), \tag{1.21}$$

where

$$T_{\mathrm{Collapse}}(P) = \frac{N_{\mathrm{Collapse}}(P)}{f} \tag{1.22}$$

and

$$T_{\mathrm{Reconstruction}}(P,\zeta) = \frac{N_{\mathrm{Reconstruction}}(P,\zeta)}{f}. \tag{1.23}$$

1.8 Runtime Gain

Definition 1.8.1 (Runtime Gain)**.** The runtime gain of DRCC relative to classical enumeration is defined by

$$G(P,\zeta) = \frac{T_{\mathrm{Classic}}(P)}{T_{\mathrm{DRCC}}(P,\zeta)}. \tag{1.24}$$

Equivalently, since both runtimes use the same reference rate f,

$$G(P,\zeta) = \frac{N_{\mathrm{Classic}}(P)}{N_{\mathrm{DRCC}}(P,\zeta)}. \tag{1.25}$$

The interpretation is:

$$G(P,\zeta) < 1 \quad \Longrightarrow \quad \text{classical enumeration is faster under the model,} \tag{1.26}$$

$$G(P,\zeta) = 1 \quad \Longrightarrow \quad \text{transition point,} \tag{1.27}$$

and

$$G(P,\zeta) > 1 \quad \Longrightarrow \quad \text{DRCC is faster under the model.} \tag{1.28}$$

Remark 1.8.2 (No Universal Speedup Claim). The condition

$$G(P,\zeta) > 1$$

is model-dependent.

It does not imply that DRCC is universally faster for all instances of a problem class.

It states only that, for the specified instance, reduction map, reconstruction space, and cost model, the DRCC runtime is smaller than the corresponding classical runtime.

1.9 Transition Point

Many DRCC examples exhibit two regimes.

For small instances, the overhead of reduction may dominate.

For larger or more structured instances, controlled reduction may become advantageous.

This motivates the transition point.

Definition 1.9.1 (Discrete Transition Point)**.** Let $n \in \mathbb{N}$ denote a discrete problem-size parameter.

Let

$$T_{\text{Classic}}(n) \tag{1.29}$$

and

$$T_{\text{DRCC}}(n) \tag{1.30}$$

be the corresponding runtime models.

A discrete transition point is a pair

$$W = (n_w, R_w[s]), \qquad n_w \in \mathbb{N}, \tag{1.31}$$

such that

$$T_{\text{Classic}}(n_w) = T_{\text{DRCC}}(n_w) = R_w[s]. \tag{1.32}$$

At the transition point,

$$G(n_w) = 1. \tag{1.33}$$

For

$$n < n_w,$$

DRCC may be slower because reduction and reconstruction overhead may exceed the cost of direct enumeration.

For

$$n > n_w,$$

DRCC may become advantageous if the reduced reconstruction space grows more slowly than the classical candidate space.

1.10 Runtime Gap

Definition 1.10.1 (Runtime Gap)**.** The runtime gap is defined by

$$\Delta_T(n) = T_{\text{Classic}}(n) - T_{\text{DRCC}}(n). \tag{1.34}$$

The sign of $\Delta_T(n)$ determines the runtime regime:

$$\Delta_T(n) < 0 \quad \Longrightarrow \quad \text{classical enumeration is faster,} \tag{1.35}$$

$$\Delta_T(n) = 0 \quad \Longrightarrow \quad \text{transition,} \tag{1.36}$$

$$\Delta_T(n) > 0 \quad \Longrightarrow \quad \text{DRCC is faster under the model.} \tag{1.37}$$

1.11 Operational Runtime Protocol

For every case study in this manuscript, the same runtime protocol is used.

Step 1. Define the classical candidate space $X(P)$.

Step 2. Compute or estimate

$$d_{\mathrm{frag}}(P) = |X(P)|.$$

Step 3. Define the controlled reduction map C_P.

Step 4. Determine the reduced state ζ.

Step 5. Construct or estimate the reconstruction space

$$\Omega_P(\zeta).$$

Step 6. Compute or estimate

$$d_{\mathrm{rec}}(P, \zeta).$$

Step 7. Define the classical evaluation count

$$N_{\mathrm{Classic}}(P).$$

Step 8. Define the DRCC evaluation count

$$N_{\mathrm{DRCC}}(P, \zeta).$$

Step 9. Compute the runtime gain

$$G(P, \zeta).$$

Step 10. Identify the transition point

$$W = (n_w, R_w[s]),$$

when applicable.

This protocol is intended to make the framework testable and comparable across examples.

1.12 Summary

This chapter introduced the core definitions used throughout the condensed manuscript.

The central quantities are:

$$d_{\mathrm{frag}}(P) = |X(P)|, \tag{1.38}$$

$$d_{\mathrm{rec}}(P, \zeta) = |\Omega_P(\zeta)|, \tag{1.39}$$

$$0 < d_{\mathrm{rec}}(P, \zeta) \leq d_{\mathrm{frag}}(P) < \infty, \tag{1.40}$$

$$T_{\mathrm{DRCC}} = T_{\mathrm{Collapse}} + T_{\mathrm{Reconstruction}}, \tag{1.41}$$

$$G(P,\zeta) = \frac{T_{\text{Classic}}(P)}{T_{\text{DRCC}}(P,\zeta)}, \tag{1.42}$$

and

$$W = (n_w, R_w[s]). \tag{1.43}$$

These definitions provide the common mathematical language for the case studies developed in later chapters.

2 Laplace-Domain View of Runtime Growth

2.1 Purpose of the Chapter

The purpose of this chapter is to introduce an analytical viewpoint on runtime growth within the DRCC framework.

The Laplace transform is not used here to solve a combinatorial problem.

It is used only as an auxiliary mathematical tool for comparing growth regimes of runtime functions.

The central objects are the classical runtime

$$T_{\text{Classic}}$$

and the DRCC runtime

$$T_{\text{DRCC}}.$$

The main question is:

> How do the runtime functions behave as the problem size increases, and when can a transition point between classical search and DRCC occur?

This chapter studies this question by considering continuous analytical approximations of discrete runtime functions.

The discrete model remains primary.

The analytical model is introduced only for growth comparison, transformability analysis, and interpretation of runtime transition.

2.2 Runtime Functions as Analytical Objects

Let

$$n \in \mathbb{N}$$

denote a discrete problem size parameter.

Examples include:

$$n = \text{number of apartments},$$

$$n = \text{number of cities},$$

$$n = \text{number of variables},$$

or

$$n = \text{number of graph vertices.}$$

In a discrete runtime model, the classical and DRCC runtimes are functions

$$T_{\text{Classic}} : \mathbb{N} \to \mathbb{R}_{\geq 0} \tag{2.1}$$

and

$$T_{\text{DRCC}} : \mathbb{N} \to \mathbb{R}_{\geq 0}. \tag{2.2}$$

For analytical comparison, one may introduce a continuous scale parameter

$$x \in \mathbb{R}_{\geq 0} \tag{2.3}$$

and continuous runtime approximations

$$T_{\text{Classic}}(x) \tag{2.4}$$

and

$$T_{\text{DRCC}}(x). \tag{2.5}$$

The continuous parameter x is not a replacement for n.
It is an analytical approximation used to study growth behavior.

2.3 Runtime Gap and Transition

The runtime gap is defined by

$$\Delta_T(x) = T_{\text{Classic}}(x) - T_{\text{DRCC}}(x). \tag{2.6}$$

The transition condition is

$$\Delta_T(x_w) = 0. \tag{2.7}$$

Equivalently,

$$T_{\text{Classic}}(x_w) = T_{\text{DRCC}}(x_w). \tag{2.8}$$

In the discrete DRCC model this transition is recorded as

$$W = (n_w, R_w[s]), \qquad n_w \in \mathbb{N}. \tag{2.9}$$

Here n_w denotes the discrete problem size at which the modeled runtime comparison changes regime, and $R_w[s]$ denotes the corresponding runtime level.

The runtime gain is

$$G(x) = \frac{T_{\text{Classic}}(x)}{T_{\text{DRCC}}(x)}. \tag{2.10}$$

The transition condition can therefore also be written as

$$G(x_w) = 1. \tag{2.11}$$

2.4 Laplace Transform of Runtime Functions

Assume that $T_{\text{Classic}}(x)$ and $T_{\text{DRCC}}(x)$ are locally integrable and satisfy suitable growth conditions.

The Laplace-domain representation of the classical runtime is

$$\mathcal{T}_{\text{Classic}}(s) = \mathcal{L}\{T_{\text{Classic}}(x)\}(s) = \int_0^\infty e^{-sx} T_{\text{Classic}}(x)\, dx. \tag{2.12}$$

Similarly, the Laplace-domain representation of the DRCC runtime is

$$\mathcal{T}_{\text{DRCC}}(s) = \mathcal{L}\{T_{\text{DRCC}}(x)\}(s) = \int_0^\infty e^{-sx} T_{\text{DRCC}}(x)\, dx. \tag{2.13}$$

The Laplace transform of the runtime gap is

$$\mathcal{D}_T(s) = \mathcal{L}\{\Delta_T(x)\}(s). \tag{2.14}$$

If both transforms in Equations 2.12 and 2.13 exist on a common half-plane, then by linearity

$$\mathcal{D}_T(s) = \mathcal{T}_{\text{Classic}}(s) - \mathcal{T}_{\text{DRCC}}(s). \tag{2.15}$$

2.5 Transformability Condition

The Laplace transform is not defined for arbitrary growth functions.

A sufficient condition for transformability is that the runtime function grows at most exponentially.

Definition 2.5.1 (Laplace-Transformable Runtime Approximation)**.** A continuous runtime approximation

$$T : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$$

is called Laplace-transformable if there exist constants

$$M > 0, \qquad \gamma \in \mathbb{R}$$

such that

$$T(x) \leq M e^{\gamma x} \qquad \text{for all sufficiently large } x. \tag{2.16}$$

Under this condition, the Laplace integral

$$\int_0^\infty e^{-sx} T(x)\, dx$$

converges for

$$\text{Re}(s) > \gamma. \tag{2.17}$$

Remark 2.5.2 (Conditional Nature of the Transform View)*.* The Laplace-domain runtime view is conditional.

If the runtime approximation is not Laplace-transformable, the transform-domain representation is not used.

In that case, the non-transformability itself indicates that the chosen continuous approximation grows too rapidly for this analytical representation.

2.6 Polynomial Runtime Growth

Polynomial runtime growth is Laplace-transformable.

Let

$$T(x) = x^m, \qquad m \in \mathbb{N}_0. \tag{2.18}$$

Then

$$\mathcal{L}\{x^m\}(s) = \int_0^\infty e^{-sx} x^m \, dx = \frac{m!}{s^{m+1}}, \qquad \mathrm{Re}(s) > 0. \tag{2.19}$$

Thus polynomial runtime functions belong to a well-controlled Laplace-transformable growth class.

In DRCC case studies, polynomial or near-polynomial DRCC runtimes are therefore analytically more controlled than exponential or factorial classical enumeration models.

2.7 Exponential Runtime Growth

Exponential runtime growth is also Laplace-transformable, but only on a shifted half-plane.

Let

$$T(x) = a^x = e^{(\log a)x}, \qquad a > 1. \tag{2.20}$$

Then

$$\mathcal{L}\{a^x\}(s) = \int_0^\infty e^{-sx} e^{(\log a)x} \, dx = \frac{1}{s - \log a}, \tag{2.21}$$

provided that

$$\mathrm{Re}(s) > \log a. \tag{2.22}$$

For Boolean enumeration, the classical candidate count is

$$2^n. \tag{2.23}$$

Its continuous approximation is

$$2^x = e^{(\log 2)x}. \tag{2.24}$$

Therefore,

$$\mathcal{L}\{2^x\}(s) = \frac{1}{s - \log 2}, \qquad \mathrm{Re}(s) > \log 2. \tag{2.25}$$

This shows that exponential classical growth is still analytically representable, but its convergence half-plane is shifted according to the exponential growth rate.

2.8 Factorial Runtime Growth

Factorial runtime growth behaves differently.

For travelling-salesman-type enumeration, the classical candidate count has the form

$$T_{\mathrm{Classic}}(n) \sim \frac{(n-1)!}{2}. \tag{2.26}$$

A natural continuous approximation is given by the Gamma function,

$$(n-1)! \rightsquigarrow \Gamma(x). \tag{2.27}$$

Using Stirling-type growth,

$$\Gamma(x+1) \sim \sqrt{2\pi x}\left(\frac{x}{e}\right)^x \qquad \text{as } x \to \infty. \tag{2.28}$$

This growth is faster than

$$e^{\gamma x}$$

for every fixed $\gamma \in \mathbb{R}$.
Consequently, the factorial approximation is not bounded by a fixed exponential function. Therefore it does not satisfy the transformability condition in Equation 2.16.

Proposition 2.8.1 (Factorial Growth Exceeds Fixed Exponential Bounds)**.** Let

$$F(x) = \Gamma(x+1).$$

Then for every fixed $\gamma \in \mathbb{R}$,

$$\frac{F(x)}{e^{\gamma x}} \to \infty \qquad \text{as } x \to \infty. \tag{2.29}$$

Proof. By Stirling-type growth,

$$\Gamma(x+1) \sim \sqrt{2\pi x}\left(\frac{x}{e}\right)^x.$$

Therefore,

$$\frac{\Gamma(x+1)}{e^{\gamma x}} \sim \sqrt{2\pi x}\left(\frac{x}{e^{\gamma+1}}\right)^x.$$

Since

$$\frac{x}{e^{\gamma+1}} \to \infty$$

as $x \to \infty$, the expression diverges to infinity.

□

Corollary 2.8.2 (Non-Transformability of Factorial Runtime Approximation)**.** The factorial runtime approximation

$$\Gamma(x+1)$$

does not satisfy the sufficient exponential-growth condition for the Laplace transform.

Proof. The result follows directly from Proposition 2.29 and the transformability condition in Equation 2.16.

□

2.9 Interpretation for DRCC

The purpose of the preceding analysis is not to claim that Laplace transforms solve combinatorial search problems.

The purpose is to classify runtime growth regimes.

Polynomial runtime models are Laplace-transformable on a right half-plane beginning at zero.

Exponential runtime models are Laplace-transformable on a shifted right half-plane.

Factorial runtime models exceed every fixed exponential bound and do not satisfy the sufficient transformability condition used in this chapter.

This distinction is useful for DRCC because many classical enumeration models grow exponentially or factorially, while DRCC attempts to replace full enumeration by controlled reduction and reconstruction.

The intended comparison is therefore:

$$T_{\text{Classic}} \quad \text{governed by full enumeration,} \tag{2.30}$$

whereas

$$T_{\text{DRCC}} = T_{\text{Collapse}} + T_{\text{Reconstruction}} \tag{2.31}$$

is governed by structural collapse and the size of the admissible reconstruction space.

The runtime transition point

$$W = (n_w, R_w[s])$$

marks the instance size at which the model predicts that controlled reconstruction becomes competitive with or advantageous over classical enumeration.

2.10 Examples of Growth Classes

The case studies later in this manuscript use the following growth classes.

Problem Type	**Classical Growth**	**Analytical Interpretation**
Housing selection	$O(nm)$	Polynomial or near-polynomial comparison.
TSG / TSP routing	$\frac{(n-1)!}{2}$	Factorial growth; not Laplace-transformable under the exponential-bound condition.
CSP	q^n	Exponential growth with rate $\log q$.
SAT	2^n	Exponential growth with rate $\log 2$.
Graph coloring	$k^{\lvert V\rvert}$	Exponential growth for fixed number of colors k.
Full Adder network	local state enumeration	Structured local collapse may reduce the effective reconstruction space.

2.11 What the Laplace View Does Not Claim

The Laplace-domain runtime view is limited.

It does not claim that a discrete runtime function must be continuous.

It does not claim that every runtime function is Laplace-transformable.

It does not claim that non-transformability is a proof of hardness.

It does not claim that DRCC solves SAT, TSP, CSP, or graph coloring in the general worst-case sense.

It provides only an analytical language for discussing growth behavior, runtime gap, and transition.

The discrete runtime model and the explicit evaluation counts remain the primary basis of comparison.

2.12 Summary

This chapter introduced a Laplace-domain viewpoint on runtime growth.

The core quantities are:

$$T_{\mathrm{Classic}}(x), \qquad T_{\mathrm{DRCC}}(x), \tag{2.32}$$

$$\Delta_T(x) = T_{\mathrm{Classic}}(x) - T_{\mathrm{DRCC}}(x), \tag{2.33}$$

$$\mathcal{T}_{\mathrm{Classic}}(s) = \mathcal{L}\{T_{\mathrm{Classic}}(x)\}, \tag{2.34}$$

$$\mathcal{T}_{\mathrm{DRCC}}(s) = \mathcal{L}\{T_{\mathrm{DRCC}}(x)\}, \tag{2.35}$$

and

$$\mathcal{D}_T(s) = \mathcal{L}\{\Delta_T(x)\}. \tag{2.36}$$

Polynomial and exponential runtime approximations are Laplace-transformable under suitable half-plane conditions.

Factorial runtime approximations exceed every fixed exponential bound and do not satisfy the sufficient transformability condition used in this chapter.

For DRCC, the main role of this viewpoint is to support the analysis of runtime growth, runtime gap, and transition behavior without replacing the underlying discrete model.

3 Algorithmic Protocol and Empirical Setup

3.1 Purpose of the Chapter

The purpose of this chapter is to make the DRCC runtime framework operational.

The previous chapters introduced the core quantities

$$d_{\mathrm{frag}}, \qquad d_{\mathrm{rec}}, \qquad T_{\mathrm{Classic}}, \qquad T_{\mathrm{DRCC}}, \qquad G, \qquad W.$$

This chapter explains how these quantities are computed or estimated in a reproducible algorithmic setting.

The objective is not to present an optimized software implementation.

The objective is to define a concrete protocol that can be applied consistently across the case studies in this manuscript.

The protocol separates three layers:

1. Classical enumeration.
2. DRCC reduction and reconstruction.
3. Runtime comparison and transition detection.

This separation is important because DRCC is not compared against an undefined notion of computation.

It is compared against a specified classical baseline and a specified DRCC reconstruction procedure.

3.2 Input Data

Let

$$P \in \mathcal{P}$$

be a finite problem instance.

The algorithmic protocol assumes the following input data:

$$P, \qquad X(P), \qquad C_P, \qquad \mathcal{A}_{\mathrm{adm}}, \qquad f.$$

Here:

P is the problem instance.

$X(P)$ is the classical candidate space.

C_P is the controlled reduction map.

$\mathcal{A}_{\mathrm{adm}}$ is the admissibility rule set.

f is the reference evaluation rate used to convert counts into wall-clock time.

The reference rate f is measured in evaluations per second.

Unless otherwise stated, count-level comparisons are independent of the particular choice of f, because the runtime gain satisfies

$$G(P,\zeta) = \frac{N_{\mathrm{Classic}}(P)}{N_{\mathrm{DRCC}}(P,\zeta)}.$$

3.3 Output Data

The protocol returns the following quantities:

$$d_{\mathrm{frag}}(P), \qquad \zeta, \qquad \Omega_P(\zeta), \qquad d_{\mathrm{rec}}(P,\zeta),$$

$$N_{\mathrm{Classic}}(P), \qquad N_{\mathrm{DRCC}}(P,\zeta),$$

$$T_{\mathrm{Classic}}(P), \qquad T_{\mathrm{DRCC}}(P,\zeta),$$

$$G(P,\zeta), \qquad W = (n_w, R_w[s]).$$

The transition point W is returned only when a parameterized family of instances is available.

For a single instance, the protocol reports the runtime gain but does not infer a transition point.

3.4 Algorithm 1: Classical Enumeration Baseline

The classical baseline evaluates every candidate in the candidate space.

Input

$$P, \qquad X(P), \qquad \mathcal{A}_{\mathrm{adm}}, \qquad f.$$

Output

$$N_{\mathrm{Classic}}(P), \qquad T_{\mathrm{Classic}}(P).$$

Protocol

Step 1. Construct or define the candidate space

$$X(P).$$

Step 2. Compute

$$d_{\mathrm{frag}}(P) = |X(P)|.$$

Step 3. For each candidate $x \in X(P)$, apply the classical admissibility or objective evaluation.

Step 4. Count the number of evaluations:

$$N_{\mathrm{Classic}}(P) = |X(P)| \cdot c_{\mathrm{eval}}(P).$$

Step 5. Convert count into runtime:

$$T_{\mathrm{Classic}}(P) = \frac{N_{\mathrm{Classic}}(P)}{f}.$$

In the unit-cost model,

$$c_{\mathrm{eval}}(P) = 1,$$

and therefore

$$N_{\mathrm{Classic}}(P) = d_{\mathrm{frag}}(P).$$

3.5 Algorithm 2: DRCC Reduction and Reconstruction

The DRCC protocol first reduces the candidate space and then reconstructs admissible candidates from the reduced representation.

Input

$$P, \qquad X(P), \qquad C_P, \qquad \mathcal{A}_{\mathrm{adm}}, \qquad f.$$

Output

$$\zeta, \qquad \Omega_P(\zeta), \qquad N_{\mathrm{DRCC}}(P, \zeta), \qquad T_{\mathrm{DRCC}}(P, \zeta).$$

Protocol

Step 1. Apply the controlled reduction map

$$C_P : X(P) \to Z(P).$$

Step 2. Determine the reduced state or structural class

$$\zeta \in Z(P).$$

Step 3. Construct the reconstruction space

$$\Omega_P(\zeta) = \{x \in X_{\text{adm}}(P) : C_P(x) = \zeta\}.$$

Step 4. Compute the reconstruction dimension

$$d_{\text{rec}}(P, \zeta) = |\Omega_P(\zeta)|.$$

Step 5. Compute the collapse count

$$N_{\text{Collapse}}(P).$$

Step 6. Compute the reconstruction count

$$N_{\text{Reconstruction}}(P, \zeta) = d_{\text{rec}}(P, \zeta) c_{\text{rec}}(P, \zeta).$$

Step 7. Compute the total DRCC count

$$N_{\text{DRCC}}(P, \zeta) = N_{\text{Collapse}}(P) + N_{\text{Reconstruction}}(P, \zeta).$$

Step 8. Convert count into runtime:

$$T_{\text{DRCC}}(P, \zeta) = \frac{N_{\text{DRCC}}(P, \zeta)}{f}.$$

Step 9. Verify structural admissibility:

$$0 < d_{\text{rec}}(P, \zeta) \leq d_{\text{frag}}(P) < \infty.$$

The final step is not optional.

If the admissibility condition fails, the reduced state is not accepted as a stable DRCC reconstruction state.

3.6 Algorithm 3: Runtime Gain and Transition Detection

For a parameterized family of problem instances

$$(P_n)_{n \in \mathbb{N}},$$

the runtime gain is computed at each problem size n.

Input

$$(P_n)_{n=n_{\min}}^{n_{\max}}, \qquad C_{P_n}, \qquad \mathcal{A}_{\text{adm}}, \qquad f.$$

Output

$$G(n), \qquad \Delta_T(n), \qquad W = (n_w, R_w[s]).$$

Protocol

Step 1. For each n, compute

$$T_{\text{Classic}}(n).$$

Step 2. For each n, compute

$$T_{\text{DRCC}}(n).$$

Step 3. Compute the runtime gap

$$\Delta_T(n) = T_{\text{Classic}}(n) - T_{\text{DRCC}}(n).$$

Step 4. Compute the runtime gain

$$G(n) = \frac{T_{\text{Classic}}(n)}{T_{\text{DRCC}}(n)}.$$

Step 5. Identify the first transition index n_w such that

$$G(n_w) \geq 1$$

and

$$G(n) < 1 \qquad \text{for at least one smaller tested value } n < n_w.$$

Step 6. Define

$$R_w[s] = T_{\text{Classic}}(n_w) = T_{\text{DRCC}}(n_w)$$

if exact equality occurs.

If exact equality does not occur in the discrete sample, report the nearest crossing interval

$$[n_-, n_+]$$

with

$$G(n_-) < 1 \qquad \text{and} \qquad G(n_+) > 1.$$

In numerical experiments, exact equality is not always expected.
For this reason, transition behavior may be reported either as a point

$$W = (n_w, R_w[s])$$

or as a crossing interval

$$W_{\text{int}} = [n_-, n_+].$$

3.7 Count-Level Validation

The first validation layer is count-level validation.

This level does not depend on implementation language, processor speed, memory hierarchy, or compiler optimizations.

It compares the number of required candidate evaluations.

The following quantities must be reported:

$$|X(P)|, \qquad |\Omega_P(\zeta)|, \qquad N_{\text{Collapse}}, \qquad N_{\text{Reconstruction}}, \qquad N_{\text{DRCC}}.$$

The count-level runtime gain is

$$G_{\text{count}}(P,\zeta) = \frac{N_{\text{Classic}}(P)}{N_{\text{DRCC}}(P,\zeta)}.$$

This is the primary validation level used in the case studies.

It is transparent and reproducible.

3.8 Wall-Clock Validation

The second validation layer is wall-clock validation.

Here one measures actual runtime on a concrete machine.

For wall-clock validation, the following data must be reported:

Field	**Required Information**
Processor	CPU or hardware platform used for the experiment.
Memory	Available memory and relevant memory limits.
Programming language	Implementation language and version.
Instance generator	How the problem instances were generated.
Classical implementation	Description of the baseline enumeration.
DRCC implementation	Description of the reduction and reconstruction procedure.
Measured quantities	Runtime, candidate counts, reconstruction counts, and gain.
Repetition protocol	Number of runs and averaging method.

Wall-clock validation is important, but it must be interpreted with care.

A poorly optimized implementation may hide the structural gain of DRCC.

Conversely, an optimized implementation may overstate practical improvement if the classical baseline is not implemented fairly.

Therefore, wall-clock experiments must always be accompanied by count-level validation.

3.9 Fair Comparison Conditions

A runtime comparison between classical enumeration and DRCC is accepted only if the following fairness conditions are satisfied.

F1. Same instance. Both methods must be applied to the same problem instance P.

F2. Same admissibility target. Both methods must aim at the same admissibility or solution criterion.

F3. Explicit baseline. The classical candidate space $X(P)$ must be explicitly defined.

F4. Explicit reduction map. The DRCC reduction map C_P must be explicitly defined.

F5. Explicit reconstruction space. The reconstruction space $\Omega_P(\zeta)$ must be explicitly defined or constructively computable.

F6. No hidden oracle. DRCC may not use information unavailable to the classical model unless that information is explicitly counted as part of the reduction cost.

F7. Counted overhead. The collapse cost N_{Collapse} must be included in N_{DRCC}.

F8. Reproducibility. The parameters and instance-generation procedure must be stated.

These conditions are included to prevent overinterpretation of runtime gain.

3.10 Empirical Evidence Table Template

Each case study in the manuscript will use a table of the following form.

Instance	N_{Classic}	N_{Collapse}	N_{Rec}	N_{DRCC}	G
P_1	–	–	–	–	–
P_2	–	–	–	–	–
P_3	–	–	–	–	–

The placeholders are replaced in the case-study chapters by the corresponding instance-specific counts.

3.11 Minimal Pseudocode

For completeness, the core DRCC runtime procedure may be written in compact pseudocode.

Input: $P,\ X(P),\ C_P,\ \mathcal{A}_{\mathrm{adm}},\ f$
Output: $T_{\mathrm{Classic}},\ T_{\mathrm{DRCC}},\ G,\ W$

1. $d_{\mathrm{frag}}(P) \leftarrow |X(P)|$
2. $N_{\mathrm{Classic}}(P) \leftarrow d_{\mathrm{frag}}(P) c_{\mathrm{eval}}(P)$
3. $\zeta \leftarrow C_P(X(P))$
4. $\Omega_P(\zeta) \leftarrow \{x \in X_{\mathrm{adm}}(P) : C_P(x) = \zeta\}$
5. $d_{\mathrm{rec}}(P, \zeta) \leftarrow |\Omega_P(\zeta)|$
6. $N_{\mathrm{DRCC}} \leftarrow N_{\mathrm{Collapse}} + d_{\mathrm{rec}}(P, \zeta) c_{\mathrm{rec}}(P, \zeta)$
7. $T_{\mathrm{Classic}} \leftarrow N_{\mathrm{Classic}}/f$
8. $T_{\mathrm{DRCC}} \leftarrow N_{\mathrm{DRCC}}/f$
9. $G \leftarrow T_{\mathrm{Classic}}/T_{\mathrm{DRCC}}$
10. accept only if $0 < d_{\mathrm{rec}}(P, \zeta) \leq d_{\mathrm{frag}}(P) < \infty$.

This pseudocode is intentionally abstract.
Each case study specifies how $X(P)$, C_P, and $\Omega_P(\zeta)$ are instantiated.

3.12 Interpretation

The algorithmic protocol clarifies the meaning of empirical evidence in this manuscript.

DRCC is not evaluated by intuition alone.

It is evaluated by:

$$N_{\text{Classic}}, \qquad N_{\text{DRCC}}, \qquad G, \qquad W.$$

The key comparison is not between two verbal descriptions.

It is between two counted procedures:

classical enumeration

and

controlled reduction plus admissible reconstruction.

This is the basis for the case studies developed in the following chapters.

3.13 Summary

This chapter introduced the algorithmic protocol used throughout the condensed manuscript.

The protocol defines:

1. a classical enumeration baseline;
2. a DRCC reduction and reconstruction procedure;
3. a runtime-gain calculation;
4. a transition-detection method;
5. count-level and wall-clock validation layers;
6. fairness conditions for runtime comparison.

The next chapters apply this protocol to representative case studies.

Each case study reports the same core quantities:

$$d_{\text{frag}}, \qquad d_{\text{rec}}, \qquad N_{\text{Classic}}, \qquad N_{\text{DRCC}}, \qquad G, \qquad W.$$

4 Housing Problem

4.1 Problem Definition

The housing problem is a multi-criteria selection problem.

A user is given a finite set of apartments

$$H = \{h_1, h_2, \ldots, h_n\}. \tag{4.1}$$

Each apartment is described by a finite list of attributes such as budget, location, size, public transport access, energy efficiency, parking, and availability.

Let

$$m \in \mathbb{N} \tag{4.2}$$

denote the number of structural criteria used for the first selection stage.

The task is to identify apartments that are compatible with the user's requirements and then select one or more admissible candidates.

The housing problem is used in this manuscript as a reviewer-oriented runtime example because it has a transparent candidate space, a natural filtering structure, and a measurable reduction from all available apartments to a smaller candidate set.

4.2 Classical Runtime Model

In the classical baseline, every apartment is evaluated against all criteria.

The classical candidate space is

$$X_{\text{Housing}}(P) = H. \tag{4.3}$$

Thus,

$$d_{\text{frag}}(P) = |H| = n. \tag{4.4}$$

Let

$$c_{\text{crit}} \tag{4.5}$$

denote the cost of checking one criterion for one apartment.

Then the structural screening cost per apartment is

$$c_{\text{screen}} = m\, c_{\text{crit}}. \tag{4.6}$$

If the classical procedure also performs detailed inspection for every apartment, let

$$c_{\text{detail}} \tag{4.7}$$

denote the cost of detailed inspection per apartment.

The classical evaluation count is then

$$N_{\text{Classic}} = n\,(c_{\text{screen}} + c_{\text{detail}})\,. \tag{4.8}$$

The corresponding runtime is

$$T_{\text{Classic}} = \frac{N_{\text{Classic}}}{f}. \tag{4.9}$$

Remark 4.2.1. If the detailed inspection cost is absent, that is if

$$c_{\text{detail}} = 0,$$

then the housing problem is only a basic filtering task.

In that case, DRCC does not automatically provide a runtime advantage.

A runtime advantage appears only when the reduction prevents expensive detail operations from being applied to all apartments.

4.3 DRCC Reduction Model

The DRCC reduction model separates the housing task into two phases.

The first phase performs structural filtering.

The second phase performs detailed reconstruction or selection only inside the reduced candidate set.

Let

$$C_{\text{Housing}} : H \to Z_{\text{Housing}} \tag{4.10}$$

be a reduction map that assigns each apartment to a structural compatibility state.

For example, an apartment may be classified according to whether it passes the budget filter, location filter, energy filter, and availability filter.

Let

$$\zeta \in Z_{\text{Housing}} \tag{4.11}$$

denote the reduced state corresponding to apartments that pass the structural filters.

The reduced candidate set is

$$\mathcal{C}_\zeta = \{h \in H : C_{\text{Housing}}(h) = \zeta\}. \tag{4.12}$$

Let

$$|\mathcal{C}_\zeta| = c \tag{4.13}$$

be the number of remaining candidates after structural reduction.

In the housing example, c is expected to be much smaller than n.

4.4 Candidate Collapse

The candidate collapse is the reduction

$$n \longrightarrow c. \tag{4.14}$$

The collapse ratio is

$$\rho_{\text{Housing}} = \frac{n}{c}. \tag{4.15}$$

If

$$\rho_{\text{Housing}} > 1, \tag{4.16}$$

then the reduction is nontrivial.

However, a nontrivial reduction alone does not imply runtime gain.

Runtime gain depends on the cost of the collapse phase, the cost of reconstruction, and the cost avoided by not applying detailed inspection to all apartments.

4.5 Reconstruction Space

The admissible reconstruction space for the housing problem is

$$\Omega_{\text{Housing}}(\zeta) = \{h \in H_{\text{adm}} : C_{\text{Housing}}(h) = \zeta\}. \tag{4.17}$$

Here,

$$H_{\text{adm}} \subseteq H \tag{4.18}$$

denotes the set of apartments that satisfy the admissibility requirements.
The reconstruction dimension is

$$d_{\text{rec}}(P, \zeta) = |\Omega_{\text{Housing}}(\zeta)|. \tag{4.19}$$

In the simplest case,

$$d_{\text{rec}}(P, \zeta) = c. \tag{4.20}$$

The structural stability condition is

$$0 < d_{\text{rec}}(P, \zeta) \leq d_{\text{frag}}(P) < \infty. \tag{4.21}$$

Substituting the housing quantities gives

$$0 < c \leq n < \infty. \tag{4.22}$$

4.6 DRCC Runtime Model

The DRCC evaluation count is decomposed into collapse cost and reconstruction cost.
The collapse cost is

$$N_{\text{Collapse}} = n\, c_{\text{screen}}. \tag{4.23}$$

The reconstruction cost is

$$N_{\text{Reconstruction}} = c\, c_{\text{detail}}. \tag{4.24}$$

Therefore,

$$N_{\text{DRCC}} = n\, c_{\text{screen}} + c\, c_{\text{detail}}. \tag{4.25}$$

The DRCC runtime is

$$T_{\text{DRCC}} = \frac{N_{\text{DRCC}}}{f}. \tag{4.26}$$

4.7 Runtime Comparison

The classical count is

$$N_{\text{Classic}} = n\,(c_{\text{screen}} + c_{\text{detail}})\,. \tag{4.27}$$

The DRCC count is

$$N_{\text{DRCC}} = n\, c_{\text{screen}} + c\, c_{\text{detail}}. \tag{4.28}$$

Subtracting gives the runtime-count gap

$$\Delta_N = N_{\text{Classic}} - N_{\text{DRCC}} \tag{4.29}$$
$$= n\,(c_{\text{screen}} + c_{\text{detail}}) - (n\,c_{\text{screen}} + c\,c_{\text{detail}}) \tag{4.30}$$
$$= (n - c)c_{\text{detail}}. \tag{4.31}$$

Therefore,

$$\Delta_N > 0 \quad \Longleftrightarrow \quad n > c \quad \text{and} \quad c_{\text{detail}} > 0. \tag{4.32}$$

This equation expresses the main runtime principle of the housing case.

DRCC becomes advantageous only when the reduced candidate set is smaller than the original set and when detailed evaluation has nonzero cost.

4.8 Runtime Gain

The runtime gain is

$$G_{\text{Housing}} = \frac{N_{\text{Classic}}}{N_{\text{DRCC}}}. \tag{4.33}$$

Using Equations 4.27 and 4.28, one obtains

$$G_{\text{Housing}} = \frac{n\,(c_{\text{screen}} + c_{\text{detail}})}{n\,c_{\text{screen}} + c\,c_{\text{detail}}}. \tag{4.34}$$

If

$$c_{\text{detail}} \gg c_{\text{screen}}$$

and

$$c \ll n,$$

then

$$G_{\text{Housing}}$$

can be significantly larger than one.

If

$$c_{\text{detail}} = 0,$$

then

$$G_{\text{Housing}} = 1. \tag{4.35}$$

Thus, DRCC does not claim automatic speedup for ordinary filtering.

The gain appears when structural reduction avoids expensive detailed inspection of candidates that are already eliminated by admissibility constraints.

4.9 Numerical Count-Level Example

Consider the following count-level example.

Let

$$n = 1200, \qquad m = 7, \qquad c = 25. \tag{4.36}$$

Assume unit criterion cost

$$c_{\text{crit}} = 1. \tag{4.37}$$

Then

$$c_{\text{screen}} = mc_{\text{crit}} = 7. \tag{4.38}$$

Assume that detailed inspection costs

$$c_{\text{detail}} = 1000 \tag{4.39}$$

operation units per apartment.
The classical count is

$$N_{\text{Classic}} = 1200(7 + 1000) \tag{4.40}$$
$$= 1,208,400. \tag{4.41}$$

The DRCC count is

$$N_{\text{DRCC}} = 1200 \cdot 7 + 25 \cdot 1000 \tag{4.42}$$
$$= 8,400 + 25,000 \tag{4.43}$$
$$= 33,400. \tag{4.44}$$

The runtime gain is

$$G_{\text{Housing}} = \frac{1,208,400}{33,400} \approx 36.18. \tag{4.45}$$

This number should not be interpreted as a universal housing speedup.
It is a count-level result under the stated cost model.

4.10 Evidence Table

Quantity	Value	Interpretation
n	1200	Number of available apartments.
m	7	Number of structural criteria.
c	25	Candidates after collapse.
d_{frag}	1200	Initial candidate size.
d_{rec}	25	Reconstruction candidate size.
N_{Classic}	$1,208,400$	Classical count under stated cost model.
N_{DRCC}	$33,400$	DRCC count under stated cost model.
G	≈ 36.18	Count-level runtime gain.

4.11 Transition Point

For the housing model, suppose that the reduced candidate count is approximately constant,

$$c(n) = c_0.$$

The classical count is

$$N_{\text{Classic}}(n) = n(c_{\text{screen}} + c_{\text{detail}}),$$

and the DRCC count is

$$N_{\text{DRCC}}(n) = nc_{\text{screen}} + c_0 c_{\text{detail}}.$$

The transition condition is

$$N_{\text{Classic}}(n_w) = N_{\text{DRCC}}(n_w). \tag{4.46}$$

Substituting the two runtime models gives

$$n_w(c_{\text{screen}} + c_{\text{detail}}) = n_w c_{\text{screen}} + c_0 c_{\text{detail}}. \tag{4.47}$$

If

$$c_{\text{detail}} > 0,$$

then

$$n_w = c_0. \tag{4.48}$$

Thus, in the idealized constant-candidate model, DRCC becomes advantageous once the number of available apartments exceeds the expected reduced candidate count.

If an additional fixed overhead

$$h > 0$$

is introduced, then

$$N_{\mathrm{DRCC}}(n) = h + nc_{\mathrm{screen}} + c_0 c_{\mathrm{detail}}.$$

The transition point becomes

$$n_w = c_0 + \frac{h}{c_{\mathrm{detail}}}. \tag{4.49}$$

This expression makes explicit that overhead delays the transition.

4.12 Laplace-Style Runtime Interpretation

For analytical comparison, introduce a continuous approximation

$$x \in \mathbb{R}_{\geq 0}$$

of the number of apartments.
In the constant-candidate model, the runtime-count functions are

$$N_{\mathrm{Classic}}(x) = x(c_{\mathrm{screen}} + c_{\mathrm{detail}}) \tag{4.50}$$

and

$$N_{\mathrm{DRCC}}(x) = xc_{\mathrm{screen}} + c_0 c_{\mathrm{detail}}. \tag{4.51}$$

Both functions are affine functions of x.
Therefore, both are Laplace-transformable on the half-plane

$$\mathrm{Re}(s) > 0.$$

One obtains

$$\mathcal{L}\{N_{\mathrm{Classic}}(x)\} = \frac{c_{\mathrm{screen}} + c_{\mathrm{detail}}}{s^2}. \tag{4.52}$$

Similarly,

$$\mathcal{L}\{N_{\mathrm{DRCC}}(x)\} = \frac{c_{\mathrm{screen}}}{s^2} + \frac{c_0 c_{\mathrm{detail}}}{s}. \tag{4.53}$$

The runtime gap is

$$\Delta_N(x) = N_{\mathrm{Classic}}(x) - N_{\mathrm{DRCC}}(x) = (x - c_0)c_{\mathrm{detail}}. \tag{4.54}$$

Thus,

$$\mathcal{L}\{\Delta_N(x)\} = c_{\mathrm{detail}} \left(\frac{1}{s^2} - \frac{c_0}{s} \right). \tag{4.55}$$

The zero of the runtime gap occurs at

$$x = c_0,$$

which agrees with the discrete transition point in Equation 4.48.

4.13 Interpretation

The housing case shows a practical form of DRCC.

The classical method evaluates every apartment in detail.

The DRCC method first applies structural filters, then performs detailed inspection only inside the reduced reconstruction space.

The key condition for runtime gain is

$$c < n$$

together with

$$c_{\text{detail}} > 0.$$

Thus, the housing example supports the following limited claim:

> For multi-stage selection problems with expensive detailed evaluation, controlled reduction can reduce runtime by restricting detailed inspection to an admissible reconstruction space.

It does not support the stronger claim that DRCC automatically improves all filtering problems.

4.14 Summary

For the housing problem, the core quantities are:

$$d_{\text{frag}} = n, \qquad d_{\text{rec}} = c. \tag{4.56}$$

The structural stability condition is

$$0 < c \leq n < \infty. \tag{4.57}$$

The runtime models are

$$N_{\text{Classic}} = n(c_{\text{screen}} + c_{\text{detail}}) \tag{4.58}$$

and

$$N_{\text{DRCC}} = nc_{\text{screen}} + cc_{\text{detail}}. \tag{4.59}$$

The runtime gain is

$$G_{\text{Housing}} = \frac{n(c_{\text{screen}} + c_{\text{detail}})}{nc_{\text{screen}} + cc_{\text{detail}}}. \tag{4.60}$$

The transition point in the constant-candidate model is

$$n_w = c. \tag{4.61}$$

The housing case therefore provides a clear example of DRCC as controlled candidate collapse followed by admissible reconstruction.

5 TSG / TSP Routing Problem

5.1 Problem Definition

Let

$$G = (V, E, w)$$

be a complete weighted graph with

$$|V| = n.$$

The vertices represent locations, and the edge-weight function

$$w : E \to \mathbb{R}_{\geq 0}$$

assigns a non-negative travel cost to every edge.

A tour is a Hamiltonian cycle visiting every vertex exactly once and returning to the starting vertex.

Let

$$\mathcal{T}_n \tag{5.1}$$

denote the set of symmetric tours, where tours are identified up to cyclic rotation and reversal.

The objective of the classical Travelling Salesman Problem is to find

$$\tau^* = \arg\min_{\tau \in \mathcal{T}_n} w(\tau), \tag{5.2}$$

where

$$w(\tau) = \sum_{e \in \tau} w(e). \tag{5.3}$$

In this manuscript, the TSG / TSP example is used to study runtime growth and structural reduction.

It is not used to claim a general solution of the TSP.

5.2 Classical Tour Enumeration

For the symmetric TSP on n labeled vertices, the number of distinct Hamiltonian cycles is

$$|\mathcal{T}_n| = \frac{(n-1)!}{2}. \tag{5.4}$$

Thus, the classical candidate space is

$$X_{\mathrm{TSP}}(P) = \mathcal{T}_n, \tag{5.5}$$

and the fragmentation dimension is

$$d_{\mathrm{frag}}(P) = |\mathcal{T}_n| = \frac{(n-1)!}{2}. \tag{5.6}$$

Under the unit-cost model, the classical evaluation count is

$$N_{\mathrm{Classic}}(n) = \frac{(n-1)!}{2}. \tag{5.7}$$

The corresponding runtime at reference evaluation rate f is

$$T_{\mathrm{Classic}}(n) = \frac{(n-1)!}{2f}. \tag{5.8}$$

This factorial growth is the main reason why exhaustive enumeration quickly becomes infeasible.

5.3 DRCC Structural Fragment

DRCC replaces direct enumeration of all tours by a structural reduction step followed by controlled reconstruction.

Let

$$F_G = (V, E_F) \tag{5.9}$$

be a sparse structural fragment of the graph.

A typical choice is a connected reference graph such as a minimum spanning tree or another admissible structural skeleton.

For the present runtime model, assume that

$$|E_F| = n - 1. \tag{5.10}$$

The fragment F_G does not itself solve the TSP.

It supplies a structural reference against which tours are classified.

5.4 DRCC Structural Classes

For a tour

$$\tau \in \mathcal{T}_n,$$

define its fragment signature by

$$C_F(\tau) = E(\tau) \cap E_F. \tag{5.11}$$

Thus,

$$C_F(\tau)$$

records which fragment edges occur in the tour.

The controlled reduction map is

$$C_F : \mathcal{T}_n \to \mathcal{Z}_F, \tag{5.12}$$

where

$$\mathcal{Z}_F \subseteq \mathfrak{P}(E_F) \tag{5.13}$$

is the set of feasible fragment signatures.

Since

$$|E_F| = n - 1,$$

one has the upper bound

$$|\mathcal{Z}_F| \leq 2^{n-1}. \tag{5.14}$$

The reduction is therefore

$$\frac{(n-1)!}{2} \longrightarrow |\mathcal{Z}_F| \leq 2^{n-1}. \tag{5.15}$$

This does not by itself solve the optimization problem.

It partitions the tour space into structural reconstruction classes.

5.5 Reconstruction Spaces

For a fragment signature

$$\zeta \in \mathcal{Z}_F,$$

the corresponding reconstruction space is

$$\Omega_F(\zeta) = \{\tau \in \mathcal{T}_n : C_F(\tau) = \zeta\}. \tag{5.16}$$

The reconstruction dimension is

$$d_{\text{rec}}(P, \zeta) = |\Omega_F(\zeta)|. \tag{5.17}$$

The structural stability condition becomes

$$0 < |\Omega_F(\zeta)| \leq |\mathcal{T}_n| < \infty. \tag{5.18}$$

This condition states that a signature is admissible only if it admits at least one tour and remains a finite subspace of the full tour space.

5.6 Controlled Reconstruction of Tours

A reconstruction procedure assigns to each feasible signature ζ one or more candidate tours.

Let

$$\mathcal{R}_F(\zeta) \subseteq \Omega_F(\zeta) \tag{5.19}$$

be the set of tours reconstructed from ζ.

Let

$$r(\zeta) = |\mathcal{R}_F(\zeta)| \tag{5.20}$$

denote the reconstruction budget for the class ζ.

The total reconstruction count is

$$N_{\text{Reconstruction}} = \sum_{\zeta \in \mathcal{Z}_F} r(\zeta). \tag{5.21}$$

If

$$r(\zeta) \leq r_{\max} \tag{5.22}$$

for every feasible signature, then

$$N_{\text{Reconstruction}} \leq |\mathcal{Z}_F|\, r_{\max} \leq 2^{n-1} r_{\max}. \tag{5.23}$$

5.7 Objective-Preserving Reconstruction

For optimization problems, a reduction must preserve enough information to recover an optimal solution.

This motivates the following condition.

Definition 5.7.1 (Objective-Preserving Reconstruction)**.** A reconstruction procedure is called objective-preserving if for every non-empty reconstruction space $\Omega_F(\zeta)$, it returns a tour

$$\tau_\zeta \in \Omega_F(\zeta) \tag{5.24}$$

satisfying

$$w(\tau_\zeta) = \min_{\tau \in \Omega_F(\zeta)} w(\tau). \tag{5.25}$$

Proposition 5.7.2 (Classwise Optimality Implies Global Optimality)**.** Assume that the feasible reconstruction spaces $\Omega_F(\zeta)$ partition $\mathcal{T}_n$.

If for every non-empty class $\Omega_F(\zeta)$ the reconstruction procedure returns a class-optimal representative τ_ζ, then

$$\min_{\zeta \in \mathcal{Z}_F} w(\tau_\zeta) = \min_{\tau \in \mathcal{T}_n} w(\tau). \tag{5.26}$$

Proof. Since the reconstruction spaces partition $\mathcal{T}_n$, every tour belongs to exactly one non-empty class $\Omega_F(\zeta)$.

For each such class, τ_ζ is assumed to minimize the tour weight inside that class.

Therefore, minimizing over all class representatives is equivalent to minimizing over all tours in the union of the classes.

Since this union is $\mathcal{T}_n$, the equality follows.

□

Remark 5.7.3. This proposition does not state that classwise optimal reconstruction is always easy.

It states only that if classwise optimal reconstruction is achieved, then global optimality is preserved.

In the runtime model, the cost of classwise reconstruction must be counted explicitly.

5.8 DRCC Runtime Model

Let

$$N_{\text{Fragment}}(n) \tag{5.27}$$

denote the cost of constructing the structural fragment F_G.

For a dense graph, a simple count-level model is

$$N_{\text{Fragment}}(n) = n^2. \tag{5.28}$$

The DRCC count is

$$N_{\text{DRCC}}(n) = N_{\text{Fragment}}(n) + N_{\text{Reconstruction}}(n). \tag{5.29}$$

Using the bounded reconstruction model,

$$N_{\text{DRCC}}(n) \leq n^2 + 2^{n-1} r_{\max}. \tag{5.30}$$

The corresponding runtime is

$$T_{\text{DRCC}}(n) = \frac{N_{\text{DRCC}}(n)}{f}. \tag{5.31}$$

5.9 Runtime Gain

The runtime gain is

$$G_{\mathrm{TSP}}(n) = \frac{N_{\mathrm{Classic}}(n)}{N_{\mathrm{DRCC}}(n)}. \tag{5.32}$$

Using the classical count and the bounded DRCC count gives the model

$$G_{\mathrm{TSP}}(n) \geq \frac{\frac{(n-1)!}{2}}{n^2 + 2^{n-1} r_{\max}}. \tag{5.33}$$

This expression shows the structural runtime mechanism.

The classical numerator grows factorially.

The DRCC denominator grows according to fragment construction and controlled reconstruction.

If $r_{\max}$ remains bounded or grows slowly, then the DRCC count grows much more slowly than classical enumeration.

5.10 Count-Level Example

The following example uses

$$r_{\max} = 3 \tag{5.34}$$

as an illustrative reconstruction budget.

The DRCC count model is

$$N_{\mathrm{DRCC}}(n) = n^2 + 3 \cdot 2^{n-1}. \tag{5.35}$$

The classical count is

$$N_{\mathrm{Classic}}(n) = \frac{(n-1)!}{2}. \tag{5.36}$$

n	N_{Classic}	N_{DRCC}	G
6	60	132	0.45
8	2520	448	5.63
10	181440	1636	110.90
12	19958400	6288	3173.41

This table is not a claim of universal TSP speedup.

It illustrates the count-level effect of replacing factorial tour enumeration by structural classes and bounded reconstruction.

5.11 Transition Point

In the illustrative model, the transition occurs when

$$\frac{(n-1)!}{2} = n^2 + 3 \cdot 2^{n-1}. \tag{5.37}$$

The table shows that

$$G(6) < 1$$

and

$$G(8) > 1.$$

Thus the discrete crossing interval is

$$W_{\text{int}} = [6, 8]. \tag{5.38}$$

In this model, DRCC overhead dominates at $n = 6$, but structural reduction becomes advantageous by $n = 8$.

5.12 Factorial Growth and Analytical Limits

The classical TSP enumeration count grows factorially:

$$N_{\text{Classic}}(n) = \frac{(n-1)!}{2}. \tag{5.39}$$

A continuous approximation uses the Gamma function:

$$(n-1)! \rightsquigarrow \Gamma(x). \tag{5.40}$$

By Stirling-type growth,

$$\Gamma(x+1)$$

exceeds every fixed exponential bound.

Therefore, this factorial approximation does not satisfy the exponential growth condition used for the Laplace-runtime view in Chapter 2.

The DRCC bound

$$n^2 + 2^{n-1} r_{\max} \tag{5.41}$$

is at most exponential when $r_{\max}$ is bounded or grows polynomially.

Thus, the analytical runtime view separates the two growth regimes:

$$\text{classical enumeration: factorial,}$$

$$\text{DRCC bounded reconstruction model: at most exponential.}$$

5.13 Interpretation

The TSG / TSP case demonstrates the strongest form of runtime contrast in this condensed manuscript.

The classical method enumerates tours.

DRCC classifies tours by structural signatures and reconstructs admissible tours inside reconstruction spaces.

The essential reduction is

$$\mathcal{T}_n \longrightarrow \mathcal{Z}_F \longrightarrow \Omega_F(\zeta). \tag{5.42}$$

The runtime advantage depends on three conditions:

1. the number of structural classes must be much smaller than the number of tours;

2. the reconstruction budget per class must remain controlled;

3. if optimality is claimed, reconstruction must preserve classwise optima.

Without the third condition, DRCC yields admissible reconstructed tours but not necessarily the globally optimal TSP tour.

5.14 Summary

For the TSG / TSP case, the classical candidate count is

$$N_{\text{Classic}}(n) = \frac{(n-1)!}{2}. \tag{5.43}$$

A DRCC structural-signature model satisfies

$$|\mathcal{Z}_F| \leq 2^{n-1}. \tag{5.44}$$

With bounded reconstruction budget $r_{\max}$,

$$N_{\text{DRCC}}(n) \leq n^2 + 2^{n-1} r_{\max}. \tag{5.45}$$

The runtime gain is

$$G_{\text{TSP}}(n) = \frac{N_{\text{Classic}}(n)}{N_{\text{DRCC}}(n)}. \tag{5.46}$$

In the illustrative example with $r_{\max} = 3$, the transition interval is

$$W_{\text{int}} = [6, 8]. \tag{5.47}$$

The TSG / TSP case therefore provides a reviewer-relevant example of how DRCC can convert factorial enumeration into a controlled reconstruction model, under explicit structural assumptions and without claiming a general solution of the TSP.

6 Constraint Satisfaction Problems

6.1 Problem Definition

A finite constraint satisfaction problem is a tuple

$$P = (\mathcal{V}, \mathcal{D}, \mathcal{C}). \tag{6.1}$$

Here

$$\mathcal{V} = \{x_1, x_2, \ldots, x_n\} \tag{6.2}$$

is a finite set of variables.
Each variable x_i has a finite domain

$$D_i. \tag{6.3}$$

For simplicity, assume in the uniform case that

$$|D_i| = q \qquad \text{for all} \qquad i = 1, \ldots, n. \tag{6.4}$$

The set

$$\mathcal{C} = \{C_1, C_2, \ldots, C_m\} \tag{6.5}$$

is a finite set of constraints.
Each constraint C_j acts on a scope

$$S_j \subseteq \mathcal{V}. \tag{6.6}$$

The arity of the constraint is

$$|S_j|. \tag{6.7}$$

Let

$$r = \max_{1 \le j \le m} |S_j| \tag{6.8}$$

be the maximum constraint arity.
The objective is to find an assignment

$$a : \mathcal{V} \to \bigcup_{i=1}^{n} D_i \tag{6.9}$$

such that

$$a(x_i) \in D_i \tag{6.10}$$

and every constraint in $\mathcal{C}$ is satisfied.

6.2 Classical Search Space

The classical candidate space consists of all complete assignments.
In the uniform-domain case,

$$X_{\mathrm{CSP}}(P) = D_1 \times D_2 \times \cdots \times D_n. \tag{6.11}$$

Therefore,

$$d_{\mathrm{frag}}(P) = |X_{\mathrm{CSP}}(P)| = q^n. \tag{6.12}$$

If each complete assignment is checked against all m constraints, then the classical evaluation count is

$$N_{\mathrm{Classic}}(P) = mq^n. \tag{6.13}$$

The corresponding runtime is

$$T_{\mathrm{Classic}}(P) = \frac{mq^n}{f}. \tag{6.14}$$

This is the exponential baseline for the CSP case study.

6.3 Constraint-Induced Collapse

DRCC does not begin by enumerating all complete assignments.

Instead, it first constructs local admissible fragments induced by the constraints.

For each constraint C_j, define the local admissible table

$$A_j = \{u \in \prod_{x_i \in S_j} D_i : C_j(u) = 1\}. \tag{6.15}$$

The table A_j contains all local assignments on the scope S_j that satisfy the constraint C_j.

Since

$$|S_j| \leq r,$$

one has

$$|A_j| \leq q^r. \tag{6.16}$$

The DRCC reduction object is the family of local admissible tables

$$\zeta = (A_1, A_2, \ldots, A_m). \tag{6.17}$$

This gives the collapse

$$q^n \longrightarrow (A_1, A_2, \ldots, A_m). \tag{6.18}$$

The cost of constructing the local tables is bounded by

$$N_{\text{Collapse}}(P) \leq mq^r. \tag{6.19}$$

This is the first DRCC reduction step.

It is local and depends on the maximum constraint arity r, not on the full number of complete assignments q^n.

6.4 Reconstruction Space

The DRCC reconstruction space consists of all complete assignments compatible with all local admissible tables.

Let

$$\pi_{S_j}(a)$$

denote the restriction of the complete assignment a to the scope S_j.

The reconstruction space is

$$\Omega_{\text{CSP}}(\zeta) = \left\{a \in X_{\text{CSP}}(P) : \pi_{S_j}(a) \in A_j \text{ for all } j = 1, \ldots, m\right\}. \tag{6.20}$$

This is exactly the set of assignments that satisfy all constraints.

The reconstruction dimension is

$$d_{\text{rec}}(P, \zeta) = |\Omega_{\text{CSP}}(\zeta)|. \tag{6.21}$$

The DRCC structural stability condition becomes

$$0 < |\Omega_{\text{CSP}}(\zeta)| \leq q^n < \infty. \tag{6.22}$$

If

$$|\Omega_{\mathrm{CSP}}(\zeta)| = 0,$$

then the CSP instance has no admissible reconstruction under the constraint system.

In that case, DRCC correctly reports an empty reconstruction space rather than a valid solution.

6.5 Controlled Reconstruction by Structural Width

To make the reconstruction phase explicit, assume that the constraint hypergraph admits a reconstruction ordering of width

$$\omega(P). \tag{6.23}$$

The width $\omega(P)$ measures the largest number of simultaneously active variables required during controlled reconstruction.

This quantity is not assumed to be small for all CSP instances.

It is an instance-dependent structural parameter.

Under a bounded-width reconstruction ordering, the number of partial states maintained during reconstruction is bounded by

$$q^{\omega(P)+1}. \tag{6.24}$$

Let

$$b(P) \tag{6.25}$$

denote the number of reconstruction steps.

Then a count-level reconstruction bound is

$$N_{\mathrm{Reconstruction}}(P, \zeta) \leq b(P) q^{\omega(P)+1}. \tag{6.26}$$

This bound is conditional.

It applies only when such a bounded-width reconstruction ordering is available.

Proposition 6.5.1 (Controlled-Width DRCC Runtime Bound)**.** Assume that a CSP instance P has maximum constraint arity r and admits a reconstruction ordering of width $\omega(P)$.

Then the DRCC evaluation count satisfies

$$N_{\mathrm{DRCC}}(P, \zeta) \leq m q^r + b(P) q^{\omega(P)+1}. \tag{6.27}$$

Proof. By Equation 6.19, the cost of constructing all local constraint tables is at most

$$m q^r.$$

By Equation 6.26, the reconstruction cost under a bounded-width ordering is at most

$$b(P) q^{\omega(P)+1}.$$

The DRCC evaluation count is the sum of collapse and reconstruction costs.

Therefore,

$$N_{\mathrm{DRCC}}(P, \zeta) \leq m q^r + b(P) q^{\omega(P)+1}.$$

□

6.6 Runtime Model

The classical evaluation count is

$$N_{\mathrm{Classic}}(P) = mq^n. \tag{6.28}$$

The DRCC evaluation count is bounded by

$$N_{\mathrm{DRCC}}(P,\zeta) \leq mq^r + b(P)q^{\omega(P)+1}. \tag{6.29}$$

The corresponding runtimes are

$$T_{\mathrm{Classic}}(P) = \frac{mq^n}{f} \tag{6.30}$$

and

$$T_{\mathrm{DRCC}}(P,\zeta) = \frac{N_{\mathrm{DRCC}}(P,\zeta)}{f}. \tag{6.31}$$

6.7 Runtime Gain

The runtime gain satisfies

$$G_{\mathrm{CSP}}(P,\zeta) = \frac{N_{\mathrm{Classic}}(P)}{N_{\mathrm{DRCC}}(P,\zeta)}. \tag{6.32}$$

Using the runtime bound, one obtains the lower estimate

$$G_{\mathrm{CSP}}(P,\zeta) \geq \frac{mq^n}{mq^r + b(P)q^{\omega(P)+1}}. \tag{6.33}$$

This expression shows the central structural condition.
DRCC becomes advantageous when

$$\omega(P) + 1$$

and

$$r$$

remain much smaller than

$$n.$$

If the structural width grows with n in an uncontrolled way, then the DRCC gain may disappear.

6.8 Numerical Count-Level Example

Consider an illustrative CSP instance family with

$$q = 3, \qquad n = 20, \qquad m = 30, \qquad r = 3, \qquad \omega(P) = 4, \qquad b(P) = 30. \tag{6.34}$$

The classical count is

$$N_{\text{Classic}} = mq^n \tag{6.35}$$
$$= 30 \cdot 3^{20} \tag{6.36}$$
$$= 104,603,532,030. \tag{6.37}$$

The collapse count is

$$N_{\text{Collapse}} = mq^r \tag{6.38}$$
$$= 30 \cdot 3^3 \tag{6.39}$$
$$= 810. \tag{6.40}$$

The reconstruction count is bounded by

$$N_{\text{Reconstruction}} \leq b(P)q^{\omega(P)+1} \tag{6.41}$$
$$= 30 \cdot 3^5 \tag{6.42}$$
$$= 7,290. \tag{6.43}$$

Therefore,

$$N_{\text{DRCC}} \leq 810 + 7,290 \tag{6.44}$$
$$= 8,100. \tag{6.45}$$

The resulting count-level gain is

$$G_{\text{CSP}} \geq \frac{104,603,532,030}{8,100} \approx 1.29 \cdot 10^7. \tag{6.46}$$

This example is conditional on the stated width model.
It is not a universal CSP speedup claim.

6.9 Evidence Table

Quantity	**Value**	**Interpretation**
q	3	Uniform domain size.
n	20	Number of variables.
m	30	Number of constraints.
r	3	Maximum constraint arity.
$\omega(P)$	4	Assumed reconstruction width.
N_{Classic}	$104,603,532,030$	Classical full-assignment count.
N_{DRCC}	$\leq 8,100$	DRCC count under width assumption.
G	$\geq 1.29 \cdot 10^7$	Conditional count-level gain.

6.10 Transition Behavior

Assume for simplicity that

$$m = b(P)$$

and that q, r, and $\omega(P)$ remain fixed.
The transition condition is

$$mq^{n_w} = mq^r + mq^{\omega(P)+1}. \tag{6.47}$$

If $m > 0$, this reduces to

$$q^{n_w} = q^r + q^{\omega(P)+1}. \tag{6.48}$$

Thus,

$$n_w = \log_q \left(q^r + q^{\omega(P)+1} \right). \tag{6.49}$$

In the numerical example,

$$q = 3, \quad r = 3, \quad \omega(P) = 4.$$

Therefore,

$$n_w = \log_3 \left(3^3 + 3^5 \right) = \log_3(270) \approx 5.10. \tag{6.50}$$

The nearest discrete transition occurs between

$$n = 5$$

and

$$n = 6.$$

6.11 Laplace-Style Runtime Interpretation

The classical CSP growth is

$$q^x = e^{(\log q)x}. \tag{6.51}$$

Its Laplace transform is

$$\mathcal{L}\{q^x\}(s) = \frac{1}{s - \log q}, \qquad \mathrm{Re}(s) > \log q. \tag{6.52}$$

Under fixed r and fixed $\omega(P)$, the DRCC count is bounded by a constant with respect to n, apart from instance-construction factors such as m and $b(P)$.

If m and $b(P)$ grow at most polynomially in n, then the DRCC runtime approximation remains polynomial or near-polynomial.

Thus, the Laplace-runtime view separates the exponential classical candidate growth

$$q^n$$

from the controlled-width DRCC reconstruction model.

6.12 Structural Conditions

The CSP case is favorable for DRCC only under structural conditions.

The following conditions are required for a meaningful runtime gain:

1. Constraint arity r must remain small relative to n.
2. The reconstruction width $\omega(P)$ must remain controlled.
3. The reconstruction ordering must be constructively available.
4. Collapse overhead must be included in the DRCC count.
5. The reconstruction space must remain admissible and non-empty.

If these conditions fail, the DRCC model may lose its advantage.

6.13 Interpretation

The CSP case study shows that DRCC is closely related to structural decomposition.

The classical method considers all complete assignments.

The DRCC method constructs local admissible tables and reconstructs global solutions through controlled compatibility.

The key chain is

$$q^n \longrightarrow (A_1, \ldots, A_m) \longrightarrow \Omega_{\mathrm{CSP}}(\zeta). \tag{6.53}$$

The runtime advantage is not automatic.

It depends on whether the structure of the constraints allows the reconstruction process to remain low-width.

6.14 Summary

For a uniform CSP with n variables, domain size q, m constraints, maximum arity r, and reconstruction width $\omega(P)$, the classical count is

$$N_{\mathrm{Classic}} = mq^n. \tag{6.54}$$

The DRCC count satisfies

$$N_{\mathrm{DRCC}} \leq mq^r + b(P)q^{\omega(P)+1}. \tag{6.55}$$

The gain estimate is

$$G_{\mathrm{CSP}} \geq \frac{mq^n}{mq^r + b(P)q^{\omega(P)+1}}. \tag{6.56}$$

The transition point under fixed structural parameters is

$$n_w = \log_q \left(q^r + q^{\omega(P)+1} \right). \tag{6.57}$$

The CSP example therefore supports a conditional DRCC claim:

> Controlled reconstruction can reduce runtime when the constraint structure admits a bounded-width reconstruction process.

It does not support a universal claim that all CSP instances become easy under DRCC.

7 Boolean Satisfiability

7.1 Problem Definition

Let

$$\Phi$$

be a Boolean formula in conjunctive normal form.
Thus,

$$\Phi = K_1 \wedge K_2 \wedge \cdots \wedge K_m, \tag{7.1}$$

where each K_j is a clause.
Let

$$\mathcal{V} = \{x_1, x_2, \ldots, x_n\} \tag{7.2}$$

be the set of Boolean variables.
Each variable has domain

$$D_i = \{0, 1\}. \tag{7.3}$$

A truth assignment is a map

$$a : \mathcal{V} \to \{0, 1\}. \tag{7.4}$$

The Boolean satisfiability problem asks whether there exists an assignment a such that

$$\Phi(a) = 1. \tag{7.5}$$

The SAT case is included because it is a canonical exponential search problem.
The purpose here is not to solve SAT in the general worst-case sense.
The purpose is to express SAT instances in the DRCC language of controlled reduction, reconstruction spaces, runtime gain, and transition behavior.

7.2 Classical Boolean Enumeration

The classical candidate space is the set of all Boolean assignments:

$$X_{\mathrm{SAT}}(\Phi) = \{0, 1\}^n. \tag{7.6}$$

Therefore,

$$d_{\mathrm{frag}}(\Phi) = |X_{\mathrm{SAT}}(\Phi)| = 2^n. \tag{7.7}$$

If every complete assignment is checked against all m clauses, then the classical evaluation count is

$$N_{\mathrm{Classic}}(\Phi) = m2^n. \tag{7.8}$$

The corresponding runtime is

$$T_{\text{Classic}}(\Phi) = \frac{m2^n}{f}. \tag{7.9}$$

This is the exponential baseline.

7.3 Clause-Induced Reduction

DRCC does not begin by enumerating all Boolean assignments.

Instead, it constructs local admissible tables induced by the clauses.

For a clause K_j, let

$$S_j \subseteq \mathcal{V} \tag{7.10}$$

be the set of variables appearing in K_j.

The clause arity is

$$|S_j|. \tag{7.11}$$

Let

$$r = \max_{1 \le j \le m} |S_j| \tag{7.12}$$

be the maximum clause width.

For each clause K_j, define the local admissible table

$$A_j = \{u \in \{0,1\}^{S_j} : K_j(u) = 1\}. \tag{7.13}$$

The reduced DRCC state is

$$\zeta = (A_1, A_2, \ldots, A_m). \tag{7.14}$$

The construction of all local tables costs at most

$$N_{\text{Collapse}}(\Phi) \le m2^r. \tag{7.15}$$

For a 3-SAT instance,

$$r = 3,$$

so each local clause table has at most

$$2^3 = 8$$

local assignments before eliminating the falsifying local pattern.

7.4 Reconstruction Space of Admissible Assignments

Let

$$\pi_{S_j}(a)$$

denote the restriction of an assignment a to the variables in S_j.

The reconstruction space associated with the reduced state ζ is

$$\Omega_{\text{SAT}}(\zeta) = \left\{a \in \{0,1\}^n : \pi_{S_j}(a) \in A_j \text{ for all } j = 1, \ldots, m\right\}. \tag{7.16}$$

Thus,

$$\Omega_{\mathrm{SAT}}(\zeta)$$

is exactly the set of satisfying assignments of Φ.

The reconstruction dimension is

$$d_{\mathrm{rec}}(\Phi, \zeta) = |\Omega_{\mathrm{SAT}}(\zeta)|. \tag{7.17}$$

If

$$d_{\mathrm{rec}}(\Phi, \zeta) > 0,$$

then the instance is satisfiable.

If

$$d_{\mathrm{rec}}(\Phi, \zeta) = 0,$$

then the reconstruction space is empty, and no satisfying assignment is compatible with the clause tables.

For the task of reconstructing a satisfying assignment, the admissible solution condition is

$$0 < d_{\mathrm{rec}}(\Phi, \zeta) \leq 2^n < \infty. \tag{7.18}$$

For the decision version of SAT, the case

$$d_{\mathrm{rec}}(\Phi, \zeta) = 0$$

is not a valid reconstruction, but it is a valid negative decision outcome if emptiness has been established by the reconstruction procedure.

7.5 Structural SAT Instances

DRCC does not assume that every SAT formula has a small reconstruction space.

A SAT instance is called structurally controlled if the reconstruction process can be performed with a bounded structural width.

Let

$$\omega(\Phi) \tag{7.19}$$

denote a reconstruction width parameter.

This parameter measures the maximum number of active variables that must be jointly maintained during controlled reconstruction.

The value of $\omega(\Phi)$ is instance-dependent.

It may be small for structured formulas and large for unstructured formulas.

If $\omega(\Phi)$ grows like n, then the DRCC advantage may disappear.

7.6 Controlled Reconstruction

Assume that a reconstruction ordering is available.

Let

$$b(\Phi) \tag{7.20}$$

denote the number of reconstruction steps.

Under a reconstruction width bound $\omega(\Phi)$, the number of active partial assignments maintained at one step is bounded by

$$2^{\omega(\Phi)+1}. \tag{7.21}$$

Thus the reconstruction count is bounded by

$$N_{\text{Reconstruction}}(\Phi, \zeta) \leq b(\Phi) 2^{\omega(\Phi)+1}. \tag{7.22}$$

The DRCC count is therefore

$$N_{\text{DRCC}}(\Phi, \zeta) \leq m2^r + b(\Phi) 2^{\omega(\Phi)+1}. \tag{7.23}$$

Proposition 7.6.1 (Controlled Structural SAT Bound)**.** Let Φ be a CNF formula with m clauses, maximum clause width r, and reconstruction width $\omega(\Phi)$.

If a reconstruction ordering with $b(\Phi)$ steps is available, then

$$N_{\text{DRCC}}(\Phi, \zeta) \leq m2^r + b(\Phi) 2^{\omega(\Phi)+1}. \tag{7.24}$$

Proof. The collapse phase constructs all local clause tables.

By Equation 7.15, this costs at most

$$m2^r.$$

The reconstruction phase maintains at most

$$2^{\omega(\Phi)+1}$$

active partial assignments per reconstruction step.

With $b(\Phi)$ steps, this gives

$$b(\Phi) 2^{\omega(\Phi)+1}.$$

Adding collapse and reconstruction costs gives the claimed bound.

□

7.7 Runtime Model

The classical count is

$$N_{\text{Classic}}(\Phi) = m2^n. \tag{7.25}$$

The DRCC count satisfies

$$N_{\text{DRCC}}(\Phi, \zeta) \leq m2^r + b(\Phi) 2^{\omega(\Phi)+1}. \tag{7.26}$$

The corresponding runtimes are

$$T_{\text{Classic}}(\Phi) = \frac{m2^n}{f} \tag{7.27}$$

and

$$T_{\text{DRCC}}(\Phi, \zeta) = \frac{N_{\text{DRCC}}(\Phi, \zeta)}{f}. \tag{7.28}$$

7.8 Runtime Gain

The runtime gain is

$$G_{\mathrm{SAT}}(\Phi,\zeta) = \frac{N_{\mathrm{Classic}}(\Phi)}{N_{\mathrm{DRCC}}(\Phi,\zeta)}. \tag{7.29}$$

Using the DRCC bound, one obtains

$$G_{\mathrm{SAT}}(\Phi,\zeta) \geq \frac{m2^n}{m2^r + b(\Phi)2^{\omega(\Phi)+1}}. \tag{7.30}$$

This expression shows the conditional nature of the runtime gain.
DRCC becomes advantageous only if

$$r$$

and

$$\omega(\Phi)$$

remain substantially smaller than

$$n.$$

7.9 Numerical Count-Level Example

Consider an illustrative structured 3-SAT family with

$$n = 40, \qquad m = 120, \qquad r = 3, \qquad \omega(\Phi) = 6, \qquad b(\Phi) = 120. \tag{7.31}$$

The classical count is

$$N_{\mathrm{Classic}} = 120 \cdot 2^{40} \tag{7.32}$$
$$= 131,941,395,333,120. \tag{7.33}$$

The collapse count is

$$N_{\mathrm{Collapse}} = 120 \cdot 2^3 \tag{7.34}$$
$$= 960. \tag{7.35}$$

The reconstruction count is bounded by

$$N_{\mathrm{Reconstruction}} \leq 120 \cdot 2^7 \tag{7.36}$$
$$= 15,360. \tag{7.37}$$

Thus,

$$N_{\mathrm{DRCC}} \leq 960 + 15,360 \tag{7.38}$$
$$= 16,320. \tag{7.39}$$

The count-level runtime gain is

$$G_{\mathrm{SAT}} \geq \frac{131,941,395,333,120}{16,320} \approx 8.08 \cdot 10^9. \tag{7.40}$$

This result is conditional on the stated reconstruction width.
It is not a worst-case SAT result.

7.10 Evidence Table

Quantity	**Value**	**Interpretation**
n	40	Number of Boolean variables.
m	120	Number of clauses.
r	3	Maximum clause width.
$\omega(\Phi)$	6	Assumed reconstruction width.
N_{Classic}	$131,941,395,333,120$	Classical full-assignment count.
N_{DRCC}	$\leq 16,320$	DRCC count under structural-width assumption.
G	$\geq 8.08 \cdot 10^9$	Conditional count-level runtime gain.

7.11 Transition Point

Assume for simplicity that

$$b(\Phi) = m$$

and that r and $\omega(\Phi)$ are fixed.
The transition condition is

$$m2^{n_w} = m2^r + m2^{\omega(\Phi)+1}. \tag{7.41}$$

If $m > 0$, this becomes

$$2^{n_w} = 2^r + 2^{\omega(\Phi)+1}. \tag{7.42}$$

Therefore,

$$n_w = \log_2\left(2^r + 2^{\omega(\Phi)+1}\right). \tag{7.43}$$

For the example values

$$r = 3, \qquad \omega(\Phi) = 6,$$

one obtains

$$n_w = \log_2(2^3 + 2^7) = \log_2(136) \approx 7.09. \tag{7.44}$$

Thus the nearest discrete transition occurs between

$$n = 7$$

and

$$n = 8.$$

7.12 Laplace-Style Runtime Interpretation

The classical SAT growth is

$$2^x = e^{(\log 2)x}. \tag{7.45}$$

Its Laplace transform is

$$\mathcal{L}\{2^x\}(s) = \frac{1}{s - \log 2}, \qquad \mathrm{Re}(s) > \log 2. \tag{7.46}$$

Under fixed structural width, the DRCC count grows according to the collapse and reconstruction terms

$$m2^r$$

and

$$b(\Phi)2^{\omega(\Phi)+1}.$$

If m and $b(\Phi)$ grow at most polynomially in n, and if r and $\omega(\Phi)$ remain controlled, then the DRCC runtime approximation remains much smaller than the classical exponential candidate enumeration.

This is a structural runtime statement, not a complexity-theoretic collapse statement.

7.13 Non-Claim: No General Proof of $begin : math : text\mathbf{P}end : math : text$ versus $begin : math : text\mathbf{NP}end : math : text$

This SAT chapter does not prove

$$P = NP$$

and does not prove

$$P \neq NP.$$

It does not claim a polynomial-time algorithm for arbitrary SAT instances.

It does not replace DPLL, CDCL, resolution, local search, or established SAT-solving methods.

The claim is narrower:

> For structurally controlled SAT instances with bounded reconstruction width, DRCC provides a count-level runtime model in which controlled local reduction and reconstruction may be significantly smaller than full Boolean enumeration.

If the reconstruction width grows with n, then the DRCC runtime advantage may disappear.

7.14 Summary

For a CNF formula with n variables and m clauses, the classical candidate count is

$$N_{\text{Classic}} = m2^n. \tag{7.47}$$

The DRCC count satisfies

$$N_{\text{DRCC}} \leq m2^r + b(\Phi)2^{\omega(\Phi)+1}. \tag{7.48}$$

The runtime gain satisfies

$$G_{\text{SAT}} \geq \frac{m2^n}{m2^r + b(\Phi)2^{\omega(\Phi)+1}}. \tag{7.49}$$

The transition point under fixed structural parameters is

$$n_w = \log_2 \left(2^r + 2^{\omega(\Phi)+1}\right). \tag{7.50}$$

The SAT case therefore supports a conditional DRCC runtime statement for structurally controlled formulas, while explicitly avoiding any claim of a general SAT solution or a resolution of the P versus NP problem.

8 Graph 3-Coloring

8.1 Problem Definition

Let

$$G = (V, E)$$

be a finite undirected graph with

$$|V| = n \qquad \text{and} \qquad |E| = m.$$

Let

$$K = \{1, 2, 3\}$$

be the set of available colors.
A coloring is a map

$$c : V \to K. \tag{8.1}$$

A coloring c is proper if

$$c(u) \neq c(v) \qquad \text{for every edge} \qquad \{u, v\} \in E. \tag{8.2}$$

The graph 3-coloring problem asks whether there exists at least one proper coloring of G.
The problem is included as a case study because it has a simple classical search space and a natural local constraint structure.

8.2 Classical Coloring Space

The classical candidate space is

$$X_{\mathrm{GC}}(G) = K^V. \tag{8.3}$$

Thus,

$$d_{\mathrm{frag}}(G) = |X_{\mathrm{GC}}(G)| = 3^n. \tag{8.4}$$

If every complete coloring is checked against every edge, the classical evaluation count is

$$N_{\mathrm{Classic}}(G) = m3^n. \tag{8.5}$$

The corresponding runtime is

$$T_{\mathrm{Classic}}(G) = \frac{m3^n}{f}. \tag{8.6}$$

This is the exponential baseline for graph coloring.

8.3 DRCC Collapse by Edge Constraints

DRCC does not begin by enumerating all colorings.

Instead, it constructs local admissible edge tables.

For each edge

$$e = \{u, v\} \in E,$$

define

$$A_e = \{(a, b) \in K^2 : a \neq b\}. \tag{8.7}$$

Since K has three colors,

$$|A_e| = 6. \tag{8.8}$$

The reduced DRCC state is the family of all local edge tables:

$$\zeta = (A_e)_{e \in E}. \tag{8.9}$$

The collapse cost is bounded by

$$N_{\mathrm{Collapse}}(G) \leq 6m. \tag{8.10}$$

This is the first reduction step:

$$3^n \longrightarrow (A_e)_{e \in E}. \tag{8.11}$$

8.4 Reconstruction Space

The reconstruction space associated with ζ is

$$\Omega_{\mathrm{GC}}(\zeta) = \left\{ c \in K^V : (c(u), c(v)) \in A_e \text{ for every } e = \{u, v\} \in E \right\}. \tag{8.12}$$

Thus,

$$\Omega_{\mathrm{GC}}(\zeta)$$

is exactly the set of proper 3-colorings of G.

The reconstruction dimension is

$$d_{\mathrm{rec}}(G, \zeta) = |\Omega_{\mathrm{GC}}(\zeta)|. \tag{8.13}$$

If

$$d_{\mathrm{rec}}(G, \zeta) > 0,$$

then G is 3-colorable.

If

$$d_{\mathrm{rec}}(G, \zeta) = 0,$$

then no admissible reconstruction exists.

For the reconstruction task, the structural stability condition is

$$0 < d_{\mathrm{rec}}(G, \zeta) \leq 3^n < \infty. \tag{8.14}$$

8.5 Reconstruction Manifold Interpretation

In earlier DRCC terminology, the reconstruction space may also be viewed as a finite reconstruction manifold or reconstruction fiber:

$$RM(G, \zeta) = \Omega_{\mathrm{GC}}(\zeta). \tag{8.15}$$

In this condensed manuscript, the term manifold is used only in the finite combinatorial sense unless additional topological or smooth structure is explicitly defined.

Thus,

$$RM(G, \zeta)$$

means the set of all colorings consistent with the reduced observation ζ.

It does not imply a differentiable manifold structure.

8.6 Controlled Reconstruction

A controlled reconstruction procedure builds colorings from the local edge tables while maintaining only a limited number of active variables.

Let

$$\omega(G) \tag{8.16}$$

denote a reconstruction width parameter.

This parameter measures the maximum number of simultaneously active vertices required during reconstruction.

Let

$$b(G) \tag{8.17}$$

denote the number of reconstruction steps.

Under a bounded-width reconstruction ordering, the number of active partial colorings is bounded by

$$3^{\omega(G)+1}. \tag{8.18}$$

Therefore,

$$N_{\text{Reconstruction}}(G, \zeta) \leq b(G)3^{\omega(G)+1}. \tag{8.19}$$

The DRCC count satisfies

$$N_{\text{DRCC}}(G, \zeta) \leq 6m + b(G)3^{\omega(G)+1}. \tag{8.20}$$

8.7 Geodesic Reconstruction Paths

If a graph structure is placed on the set of partial reconstructions, then a reconstruction path is a sequence

$$\gamma = (c_0, c_1, \ldots, c_\ell) \tag{8.21}$$

where each c_i is a partial coloring and each step adds or modifies one admissible local decision.

The length of the path is

$$\text{length}(\gamma) = \ell. \tag{8.22}$$

A geodesic reconstruction path is a shortest admissible path from an initial partial state to a proper coloring.

Formally,

$$GP(G, \zeta, c) = \arg\min_{\gamma} \text{length}(\gamma), \tag{8.23}$$

where γ ranges over admissible paths ending at

$$c \in \Omega_{\text{GC}}(\zeta).$$

This path viewpoint measures reconstruction cost inside the reconstruction space.

It is not required for every runtime estimate, but it provides a geometric interpretation of controlled reconstruction.

8.8 Runtime Gain

The runtime gain is

$$G_{\text{GC}}(G, \zeta) = \frac{N_{\text{Classic}}(G)}{N_{\text{DRCC}}(G, \zeta)}. \tag{8.24}$$

Using the runtime bound gives

$$G_{\mathrm{GC}}(G,\zeta) \geq \frac{m3^n}{6m + b(G)3^{\omega(G)+1}}. \tag{8.25}$$

The gain is significant only when

$$\omega(G) \ll n.$$

If $\omega(G)$ grows with n in an uncontrolled way, then the DRCC advantage may disappear.

8.9 Numerical Count-Level Example

Consider an illustrative graph with

$$n = 24, \qquad m = 36, \qquad \omega(G) = 4, \qquad b(G) = 24. \tag{8.26}$$

The classical count is

$$N_{\mathrm{Classic}} = 36 \cdot 3^{24} \tag{8.27}$$

$$= 10,167,463,313,316. \tag{8.28}$$

The collapse count is

$$N_{\mathrm{Collapse}} = 6m = 216. \tag{8.29}$$

The reconstruction count is bounded by

$$N_{\mathrm{Reconstruction}} \leq 24 \cdot 3^5 \tag{8.30}$$

$$= 5,832. \tag{8.31}$$

Thus,

$$N_{\mathrm{DRCC}} \leq 6,048. \tag{8.32}$$

The count-level gain is

$$G_{\mathrm{GC}} \geq \frac{10,167,463,313,316}{6,048} \approx 1.68 \cdot 10^9. \tag{8.33}$$

This number is conditional on the assumed reconstruction width.
It is not a universal graph-coloring speedup.

8.10 Evidence Table

Quantity	**Value**	**Interpretation**
n	24	Number of vertices.
m	36	Number of edges.
$\omega(G)$	4	Assumed reconstruction width.
d_{frag}	3^{24}	Classical coloring space.
N_{Classic}	$10,167,463,313,316$	Classical edge-check count.
N_{DRCC}	$\leq 6,048$	DRCC count under width assumption.
G	$\geq 1.68 \cdot 10^9$	Conditional count-level gain.

8.11 Transition Point

Assume for simplicity that $b(G)$ and m scale proportionally and that $\omega(G)$ remains fixed.

The transition condition is

$$m3^{n_w} = 6m + b(G)3^{\omega(G)+1}. \tag{8.34}$$

If $m > 0$, then

$$3^{n_w} = 6 + \frac{b(G)}{m}3^{\omega(G)+1}. \tag{8.35}$$

Thus,

$$n_w = \log_3\left(6 + \frac{b(G)}{m}3^{\omega(G)+1}\right). \tag{8.36}$$

For the numerical example,

$$\frac{b(G)}{m} = \frac{24}{36} = \frac{2}{3},$$

and

$$\omega(G) = 4.$$

Hence,

$$n_w = \log_3\left(6 + \frac{2}{3}3^5\right) = \log_3(168) \approx 4.66. \tag{8.37}$$

The nearest discrete transition is therefore around

$$n = 5.$$

8.12 Interpretation

The graph-coloring case illustrates a finite reconstruction-space viewpoint.

Classical search enumerates all colorings in

$$K^V.$$

DRCC constructs local admissibility tables for edges and reconstructs proper colorings from compatible local decisions.

The core chain is

$$3^n \longrightarrow (A_e)_{e\in E} \longrightarrow \Omega_{\mathrm{GC}}(\zeta). \tag{8.38}$$

The runtime advantage depends on structural width.

Thus, the graph-coloring example supports a conditional claim:

> Controlled reconstruction can reduce enumeration cost for graph-coloring instances whose constraint structure admits bounded-width reconstruction.

It does not imply that arbitrary graph coloring becomes easy.

9 Full Adder

9.1 Problem Definition

The full adder is a finite Boolean reconstruction problem.

Let

$$B = \{0, 1\}$$

be the Boolean alphabet.
The input space of the full adder is

$$X_{\mathrm{FA}} = B^3. \tag{9.1}$$

An element of X_{FA} is written as

$$x = (a, b, c_{\mathrm{in}}),$$

where a and b are the two input bits and c_{in} is the carry-in bit.
Thus,

$$|X_{\mathrm{FA}}| = 2^3 = 8. \tag{9.2}$$

The full adder computes two output bits:

$$s \qquad \text{and} \qquad c_{\mathrm{out}},$$

where s is the sum bit and c_{out} is the carry-out bit.
The full adder map is

$$F_{\mathrm{FA}} : B^3 \to B^2. \tag{9.3}$$

It is defined by

$$F_{\mathrm{FA}}(a, b, c_{\mathrm{in}}) = (s, c_{\mathrm{out}}). \tag{9.4}$$

The output bits are given by

$$s = a \oplus b \oplus c_{\mathrm{in}}, \tag{9.5}$$

and

$$c_{\mathrm{out}} = ab \vee ac_{\mathrm{in}} \vee bc_{\mathrm{in}}. \tag{9.6}$$

Equivalently, if

$$h(x) = a + b + c_{\mathrm{in}}, \tag{9.7}$$

then

$$s = h(x) \bmod 2 \tag{9.8}$$

and

$$c_{\text{out}} = \begin{cases} 0, & h(x) \in \{0,1\}, \\ 1, & h(x) \in \{2,3\}. \end{cases} \tag{9.9}$$

The full adder is included as a micro-case because it is finite, complete, reproducible, and small enough to show the entire DRCC collapse explicitly.

9.2 Classical Candidate Space

The classical candidate space is the full Boolean cube

$$X^{\text{FA}}_{\text{Classic}} = B^3. \tag{9.10}$$

Therefore,

$$d_{\text{frag}}(X_{\text{FA}}) = |X^{\text{FA}}_{\text{Classic}}| = 8. \tag{9.11}$$

If each input vector is evaluated separately, the classical evaluation count is

$$N^{\text{FA}}_{\text{Classic}} = 8. \tag{9.12}$$

The corresponding runtime is

$$T^{\text{FA}}_{\text{Classic}} = \frac{8}{f}, \tag{9.13}$$

where f denotes the evaluation frequency or elementary operation rate.

This is not a large runtime. The purpose of the full adder case is not to demonstrate a large speedup, but to show a complete controlled collapse in a transparent finite setting.

9.3 DRCC Collapse by Hamming Weight

DRCC does not treat all eight Boolean inputs as structurally distinct.

The relevant structural quantity is the Hamming weight

$$h : B^3 \to \{0,1,2,3\}, \tag{9.14}$$

defined by

$$h(a,b,c_{\text{in}}) = a + b + c_{\text{in}}. \tag{9.15}$$

The controlled reduction map is

$$R_{\text{FA}} : B^3 \to \{0,1,2,3\}. \tag{9.16}$$

It is defined by

$$R_{\text{FA}}(x) = h(x). \tag{9.17}$$

Thus, the eight Boolean input states collapse into four structural classes.

For each $k \in \{0,1,2,3\}$, define

$$C_k = \{x \in B^3 : h(x) = k\}. \tag{9.18}$$

The family of structural classes is

$$\mathcal{C}_{\text{FA}} = \{C_0, C_1, C_2, C_3\}. \tag{9.19}$$

The class sizes are

$$|C_0| = 1, \qquad |C_1| = 3, \qquad |C_2| = 3, \qquad |C_3| = 1. \tag{9.20}$$

The DRCC reduced state is

$$\zeta_{\mathrm{FA}} = \mathcal{C}_{\mathrm{FA}}. \tag{9.21}$$

The collapse chain is

$$8 \longrightarrow 4. \tag{9.22}$$

Equivalently,

$$B^3 \longrightarrow \{C_0, C_1, C_2, C_3\}. \tag{9.23}$$

The collapse count is bounded by the number of input states inspected:

$$N^{\mathrm{FA}}_{\mathrm{Collapse}} \leq 8. \tag{9.24}$$

If the Hamming weight is computed directly from the three bits, the elementary bit-addition count is bounded by

$$N^{\mathrm{FA}}_{\mathrm{Collapse,bit}} \leq 3 \cdot 8 = 24. \tag{9.25}$$

The first count measures state inspection. The second count measures a more explicit bit-level implementation.

9.4 Output Reconstruction from Structural Classes

The output of the full adder depends only on the Hamming weight.

Define the class-output map

$$Q_{\mathrm{FA}} : \{0, 1, 2, 3\} \to B^2. \tag{9.26}$$

It is defined by

$$Q_{\mathrm{FA}}(k) = (k \bmod 2, \mathbf{1}_{\{k \geq 2\}}), \tag{9.27}$$

where $\mathbf{1}_{\{k \geq 2\}}$ denotes the indicator function of the condition $k \geq 2$.

Thus,

$$F_{\mathrm{FA}} = Q_{\mathrm{FA}} \circ R_{\mathrm{FA}}. \tag{9.28}$$

Equation (9.28) is the central DRCC factorization for the full adder case.

It says that the full adder output can be reconstructed from the collapsed Hamming-weight class.

The explicit class-output table is

Class	**Hamming weight**	**Sum bit**	**Carry-out bit**
C_0	0	0	0
C_1	1	1	0
C_2	2	0	1
C_3	3	1	1

This table is the finite reconstruction table of the full adder.

9.5 Reconstruction Space

For a reduced class value

$$k \in \{0, 1, 2, 3\},$$

the reconstruction fiber is

$$\Omega_{\mathrm{FA}}(k) = \{x \in B^3 : R_{\mathrm{FA}}(x) = k\}. \tag{9.29}$$

Since $R_{\mathrm{FA}}(x) = h(x)$, this can also be written as

$$\Omega_{\mathrm{FA}}(k) = C_k. \tag{9.30}$$

The reconstruction dimension of a class is

$$d_{\mathrm{rec}}^{\mathrm{FA}}(k) = |\Omega_{\mathrm{FA}}(k)|. \tag{9.31}$$

Therefore,

$$d_{\mathrm{rec}}^{\mathrm{FA}}(0) = 1, \qquad d_{\mathrm{rec}}^{\mathrm{FA}}(1) = 3, \qquad d_{\mathrm{rec}}^{\mathrm{FA}}(2) = 3, \qquad d_{\mathrm{rec}}^{\mathrm{FA}}(3) = 1. \tag{9.32}$$

The total reduced structural dimension is

$$R_{\mathrm{DRCC}}^{\mathrm{FA}} = |\mathcal{C}_{\mathrm{FA}}| = 4. \tag{9.33}$$

The stability condition for each nonempty reconstruction fiber is

$$0 < d_{\mathrm{rec}}^{\mathrm{FA}}(k) \leq d_{\mathrm{frag}}(X_{\mathrm{FA}}) = 8 < \infty. \tag{9.34}$$

Since every class C_k is nonempty, the full adder satisfies the finite DRCC stability condition for all four structural classes.

9.6 Controlled Reconstruction Operator

The controlled reconstruction operator acts on a reduced class value and returns the corresponding output pair.

Define

$$\mathcal{R}_{\mathrm{FA}} : \{0, 1, 2, 3\} \to B^2. \tag{9.35}$$

It is defined by

$$\mathcal{R}_{\mathrm{FA}}(k) = Q_{\mathrm{FA}}(k). \tag{9.36}$$

Thus,

$$\mathcal{R}_{\mathrm{FA}}(0) = (0, 0), \tag{9.37}$$

$$\mathcal{R}_{\mathrm{FA}}(1) = (1, 0), \tag{9.38}$$

$$\mathcal{R}_{\mathrm{FA}}(2) = (0, 1), \tag{9.39}$$

and

$$\mathcal{R}_{\mathrm{FA}}(3) = (1, 1). \tag{9.40}$$

For every input $x \in B^3$, the original full adder output is recovered by

$$F_{\mathrm{FA}}(x) = \mathcal{R}_{\mathrm{FA}}(R_{\mathrm{FA}}(x)). \tag{9.41}$$

This identity shows that the reduction is lossless with respect to the full adder output.

It does not preserve the identity of the original input state inside a class. It preserves only the information required to reconstruct the output.

9.7 Algorithmic Protocol

The full adder case admits a direct DRCC-style protocol.

Step	**Operation**
1	Read the input state $x = (a, b, c_{\mathrm{in}}) \in B^3$.
2	Compute the structural class $k = R_{\mathrm{FA}}(x) = a + b + c_{\mathrm{in}}$.
3	Use the reconstruction operator $\mathcal{R}_{\mathrm{FA}}(k)$.
4	Return the output pair $(s, c_{\mathrm{out}}) = \mathcal{R}_{\mathrm{FA}}(k)$.

Equivalently, the protocol can be written as the following finite pseudocode.

```
Input:  a, b, c_in in {0,1}

k := a + b + c_in

if k = 0 then return (0,0)
if k = 1 then return (1,0)
if k = 2 then return (0,1)
if k = 3 then return (1,1)
```

This protocol is fully reproducible because all input states, reduction classes, and output states are finite and explicitly defined.

9.8 Runtime Count

The DRCC runtime count separates collapse and reconstruction.

The collapse step computes

$$k = a + b + c_{\mathrm{in}}.$$

This requires a constant number of elementary operations.

Thus,

$$N^{\mathrm{FA}}_{\mathrm{Collapse}}(x) \leq 3. \tag{9.42}$$

The reconstruction step performs one table lookup or one class-output evaluation:

$$N^{\mathrm{FA}}_{\mathrm{Reconstruction}}(x) \leq 1. \tag{9.43}$$

Therefore, for one input state,

$$N^{\mathrm{FA}}_{\mathrm{DRCC}}(x) \leq 4. \tag{9.44}$$

For all eight input states, the complete count is bounded by

$$N_{\mathrm{DRCC}}^{\mathrm{FA}} \leq 8 \cdot 4 = 32. \tag{9.45}$$

This count is an implementation-level bound. It is not directly smaller than the symbolic classical state count 8, because it counts different elementary operations.

For a fair structural comparison, DRCC counts the number of distinct output-relevant classes rather than the number of raw inputs.

The structural DRCC count is

$$N_{\mathrm{DRCC,struct}}^{\mathrm{FA}} = 4. \tag{9.46}$$

The corresponding structural gain is

$$G_{\mathrm{FA,struct}} = \frac{N_{\mathrm{Classic}}^{\mathrm{FA}}}{N_{\mathrm{DRCC,struct}}^{\mathrm{FA}}} = \frac{8}{4} = 2. \tag{9.47}$$

This is a structural reduction factor, not a claim of hardware-level speedup.

9.9 Evidence Table

Quantity	**Value**	**Interpretation**
Input space	B^3	All Boolean input triples.
$\lvert X_{\mathrm{FA}}\rvert$	8	Classical candidate states.
Reduction map	$R_{\mathrm{FA}}(x) = h(x)$	Collapse by Hamming weight.
Structural classes	C_0, C_1, C_2, C_3	Four output-relevant classes.
$R_{\mathrm{DRCC}}^{\mathrm{FA}}$	4	Number of distinct structural classes.
d_{frag}	8	Fragment space before collapse.
$d_{\mathrm{rec}}(k)$	$1, 3, 3, 1$	Fiber sizes after collapse.
$G_{\mathrm{FA,struct}}$	2	Structural reduction factor.

9.10 Transition Interpretation

For the full adder, the transition point is not asymptotic.

The input size is fixed at three Boolean variables.

Thus, the transition point is interpreted as a finite structural threshold:

$$|X_{\mathrm{FA}}| = 8 \qquad \text{and} \qquad |\mathcal{C}_{\mathrm{FA}}| = 4. \tag{9.48}$$

The finite transition condition is

$$|\mathcal{C}_{\mathrm{FA}}| < |X_{\mathrm{FA}}|. \tag{9.49}$$

Since

$$4 < 8, \tag{9.50}$$

the full adder admits a nontrivial output-preserving collapse.

This transition is not a runtime phase transition in the asymptotic sense.

It is a finite demonstration that output-relevant information may have smaller structural dimension than the raw input space.

9.11 Interpretation

The full adder case illustrates the simplest complete DRCC pattern.

The classical view distinguishes all eight Boolean input states:

$$B^3.$$

The DRCC view collapses these states into four Hamming-weight classes:

$$C_0, C_1, C_2, C_3.$$

The output is reconstructed from the class value rather than from the identity of the individual input state.

The central chain is

$$B^3 \longrightarrow \{0, 1, 2, 3\} \longrightarrow B^2. \tag{9.51}$$

Equivalently,

$$x \longrightarrow R_{\mathrm{FA}}(x) \longrightarrow \mathcal{R}_{\mathrm{FA}}(R_{\mathrm{FA}}(x)). \tag{9.52}$$

This demonstrates the DRCC principle in a finite and reproducible form:

> Preserve the information required for reconstruction and collapse the input distinctions that are irrelevant to the required output.

For the full adder, the collapsed Hamming-weight class is sufficient to reconstruct the output pair.

It is not sufficient to reconstruct the original input uniquely.

Thus, the reduction is output-preserving but not input-invertible.

9.12 Limit of the Full Adder Case

The full adder is a micro-case.

It does not establish a general complexity-theoretic result.

It does not imply that arbitrary Boolean circuits admit a comparable collapse.

It only shows that, for this specific finite Boolean reconstruction problem, the output-relevant structure is smaller than the full input space.

This limitation is important because DRCC-V2 does not claim a universal algorithmic speedup.

The full adder provides a controlled example of structural compression, not a proof of broad computational tractability.

9.13 Summary

For the full adder, the classical candidate space is

$$X_{\mathrm{FA}} = B^3. \tag{9.53}$$

The fragment dimension is

$$d_{\mathrm{frag}}(X_{\mathrm{FA}}) = 8. \tag{9.54}$$

The controlled reduction map is

$$R_{\mathrm{FA}}(x) = h(x). \tag{9.55}$$

The reduced structural class space is

$$\mathcal{C}_{\mathrm{FA}} = \{C_0, C_1, C_2, C_3\}. \tag{9.56}$$

The number of DRCC structural classes is

$$R_{\mathrm{DRCC}}^{\mathrm{FA}} = 4. \tag{9.57}$$

The reconstruction identity is

$$F_{\mathrm{FA}}(x) = \mathcal{R}_{\mathrm{FA}}(R_{\mathrm{FA}}(x)). \tag{9.58}$$

The structural gain is

$$G_{\mathrm{FA,struct}} = 2. \tag{9.59}$$

The full adder case therefore gives a complete finite example of controlled collapse, reconstruction by a reduced class, and explicit runtime accounting.

10 Cross-Case Runtime Analysis

10.1 Purpose of the Cross-Case Analysis

This chapter compares the runtime behavior of the case studies developed in the preceding chapters.

The purpose is not to prove a universal speedup theorem.

The purpose is to show, in a common notation, how the DRCC runtime model behaves across different finite reconstruction settings.

The analysis follows the operational runtime protocol introduced in Chapter 1.

The central quantities are:

$$N_{\mathrm{Classic}}, \qquad N_{\mathrm{Collapse}}, \qquad N_{\mathrm{Reconstruction}}, \qquad N_{\mathrm{DRCC}}, \qquad G, \qquad W.$$

The comparison is count-level.

Wall-clock runtime depends on implementation, hardware, memory access, data representation, and benchmark selection.

Thus, the following analysis should be read as a structural runtime comparison, not as an empirical benchmark.

10.2 Common Runtime Decomposition

Let

$$P \in \mathcal{P}$$

be a finite problem instance.

The classical evaluation count is

$$N_{\mathrm{Classic}}(P) = |X(P)| \cdot c_{\mathrm{eval}}(P). \tag{10.1}$$

The DRCC evaluation count is decomposed as

$$N_{\mathrm{DRCC}}(P,\zeta) = N_{\mathrm{Collapse}}(P) + N_{\mathrm{Reconstruction}}(P,\zeta). \tag{10.2}$$

The runtime gain is

$$G(P,\zeta) = \frac{N_{\mathrm{Classic}}(P)}{N_{\mathrm{DRCC}}(P,\zeta)}. \tag{10.3}$$

The DRCC regime is favorable under the model when

$$G(P,\zeta) > 1. \tag{10.4}$$

Equivalently,

$$N_{\mathrm{Classic}}(P) > N_{\mathrm{DRCC}}(P,\zeta). \tag{10.5}$$

Using the decomposition in Equation (10.2), this becomes

$$N_{\mathrm{Classic}}(P) > N_{\mathrm{Collapse}}(P) + N_{\mathrm{Reconstruction}}(P,\zeta). \tag{10.6}$$

This inequality is the common runtime criterion used in the comparison below.

10.3 Structural Growth Types

The case studies in this manuscript exhibit different growth patterns.

For a problem-size parameter n, the classical candidate space often has one of the following forms:

$$|X(P_n)| = a^n, \qquad a > 1, \tag{10.7}$$

or

$$|X(P_n)| = n!, \tag{10.8}$$

or a finite constant size in micro-cases.

DRCC becomes potentially advantageous when the reduced reconstruction cost grows more slowly than the classical candidate space.

A typical sufficient count-level pattern is

$$N_{\mathrm{Collapse}}(P_n) + N_{\mathrm{Reconstruction}}(P_n,\zeta_n) \ll N_{\mathrm{Classic}}(P_n). \tag{10.9}$$

This condition is not automatic.

It depends on the existence of a useful controlled reduction map and a bounded or slowly growing reconstruction space.

10.4 Case: Graph 3-Coloring

For graph 3-coloring, let

$$G = (V,E), \qquad |V| = n, \qquad |E| = m.$$

The classical candidate space is

$$X_{\mathrm{GC}}(G) = K^V, \qquad K = \{1,2,3\}. \tag{10.10}$$

Thus,

$$d_{\text{frag}}(G) = 3^n. \tag{10.11}$$

The classical evaluation count used in Chapter 8 is

$$N^{\text{GC}}_{\text{Classic}}(G) = m3^n. \tag{10.12}$$

The DRCC count satisfies

$$N^{\text{GC}}_{\text{DRCC}}(G, \zeta) \leq 6m + b(G)3^{\omega(G)+1}. \tag{10.13}$$

The corresponding gain satisfies

$$G_{\text{GC}}(G, \zeta) \geq \frac{m3^n}{6m + b(G)3^{\omega(G)+1}}. \tag{10.14}$$

The decisive structural parameter is

$$\omega(G).$$

If $\omega(G) \ll n$, then reconstruction is controlled by a much smaller active-width parameter.
If $\omega(G)$ grows like n, the advantage may disappear.
Thus, graph coloring supports only a conditional runtime claim:

> Controlled reconstruction may reduce enumeration cost for graph-coloring instances that admit bounded-width or slowly growing-width reconstruction.

10.5 Case: Full Adder

For the full adder, the input space is

$$X_{\text{FA}} = B^3, \qquad B = \{0, 1\}. \tag{10.15}$$

Hence,

$$d_{\text{frag}}(X_{\text{FA}}) = 8. \tag{10.16}$$

The controlled reduction map is the Hamming-weight map

$$R_{\text{FA}}(x) = h(x). \tag{10.17}$$

The reduced structural class space is

$$\mathcal{C}_{\text{FA}} = \{C_0, C_1, C_2, C_3\}. \tag{10.18}$$

Thus,

$$R^{\text{FA}}_{\text{DRCC}} = 4. \tag{10.19}$$

The structural gain is

$$G_{\text{FA,struct}} = \frac{8}{4} = 2. \tag{10.20}$$

This is not an asymptotic result.

It is a finite micro-case showing that output-relevant information may have smaller structural dimension than the full input space.

The full adder therefore serves as a complete reproducibility example, not as evidence of general complexity reduction for arbitrary Boolean circuits.

10.6 Case: Constraint Satisfaction Problems

Let a constraint satisfaction problem be given by

$$P_{\mathrm{CSP}} = (V, D, \mathcal{C}),$$

where V is a finite variable set, D is a finite domain, and $\mathcal{C}$ is a finite family of constraints.
Let

$$|V| = n \qquad \text{and} \qquad |D| = q.$$

The classical candidate space is

$$X_{\mathrm{CSP}}(P) = D^V. \tag{10.21}$$

Thus,

$$d_{\mathrm{frag}}(P_{\mathrm{CSP}}) = q^n. \tag{10.22}$$

If every full assignment is checked against all constraints, a simple classical count is

$$N^{\mathrm{CSP}}_{\mathrm{Classic}}(P) = |\mathcal{C}|q^n. \tag{10.23}$$

A DRCC-style reduction groups assignments or partial assignments by constraint-relevant structural states.
Let

$$C_P : D^V \to Z(P) \tag{10.24}$$

be the controlled reduction map.
For a reduced state $\zeta \in Z(P)$, the reconstruction space is

$$\Omega_{\mathrm{CSP}}(\zeta) = \{x \in X_{\mathrm{adm}}(P) : C_P(x) = \zeta\}. \tag{10.25}$$

The DRCC evaluation count is

$$N^{\mathrm{CSP}}_{\mathrm{DRCC}}(P, \zeta) = N^{\mathrm{CSP}}_{\mathrm{Collapse}}(P) + N^{\mathrm{CSP}}_{\mathrm{Reconstruction}}(P, \zeta). \tag{10.26}$$

The gain is

$$G_{\mathrm{CSP}}(P, \zeta) = \frac{|\mathcal{C}|q^n}{N^{\mathrm{CSP}}_{\mathrm{Collapse}}(P) + N^{\mathrm{CSP}}_{\mathrm{Reconstruction}}(P, \zeta)}. \tag{10.27}$$

This expression is useful only after the reduction map and reconstruction procedure have been specified.

Therefore, the CSP case is a framework-level runtime model unless a concrete constraint family and implementation are fixed.

10.7 Case: SAT

Let a Boolean satisfiability instance be

$$\Phi = C_1 \wedge C_2 \wedge \cdots \wedge C_m, \tag{10.28}$$

over variables

$$x_1, \ldots, x_n.$$

The classical candidate space is

$$X_{\mathrm{SAT}}(\Phi) = \{0,1\}^n. \tag{10.29}$$

Thus,

$$d_{\mathrm{frag}}(\Phi) = 2^n. \tag{10.30}$$

If each assignment is checked against all m clauses, the classical evaluation count is

$$N^{\mathrm{SAT}}_{\mathrm{Classic}}(\Phi) = m2^n. \tag{10.31}$$

A DRCC reduction may group assignments by clause-response vectors.
Let

$$C_\Phi : \{0,1\}^n \to \{0,1\}^m \tag{10.32}$$

be defined by

$$C_\Phi(x) = (C_1(x), C_2(x), \ldots, C_m(x)). \tag{10.33}$$

The satisfying class corresponds to the vector

$$\mathbf{1}_m = (1,1,\ldots,1). \tag{10.34}$$

The reconstruction space for satisfiable assignments is

$$\Omega_{\mathrm{SAT}}(\mathbf{1}_m) = \{x \in \{0,1\}^n : C_\Phi(x) = \mathbf{1}_m\}. \tag{10.35}$$

The reconstruction dimension is

$$d_{\mathrm{rec}}(\Phi, \mathbf{1}_m) = |\Omega_{\mathrm{SAT}}(\mathbf{1}_m)|. \tag{10.36}$$

The DRCC count is

$$N^{\mathrm{SAT}}_{\mathrm{DRCC}}(\Phi, \mathbf{1}_m) = N^{\mathrm{SAT}}_{\mathrm{Collapse}}(\Phi) + N^{\mathrm{SAT}}_{\mathrm{Reconstruction}}(\Phi, \mathbf{1}_m). \tag{10.37}$$

The gain is

$$G_{\mathrm{SAT}}(\Phi, \mathbf{1}_m) = \frac{m2^n}{N^{\mathrm{SAT}}_{\mathrm{Collapse}}(\Phi) + N^{\mathrm{SAT}}_{\mathrm{Reconstruction}}(\Phi, \mathbf{1}_m)}. \tag{10.38}$$

This does not imply a polynomial-time SAT algorithm.

It states only that a specified SAT instance may exhibit useful structural grouping if the clause-vector collapse and reconstruction procedure are cheaper than exhaustive assignment evaluation.

10.8 Case: TSG/TSP

For a travelling-salesman-type routing problem, let

$$V = \{v_1, \ldots, v_n\}$$

be the set of cities or nodes.
The classical candidate space consists of permutations of V.
Thus,

$$X_{\mathrm{TSP}}(P) = S_n, \tag{10.39}$$

where S_n is the symmetric group on n elements.
Therefore,

$$d_{\mathrm{frag}}(P_{\mathrm{TSP}}) = |S_n| = n!. \tag{10.40}$$

If each route is evaluated once, then

$$N^{\mathrm{TSP}}_{\mathrm{Classic}}(P) = n!. \tag{10.41}$$

If route evaluation includes edge-cost accumulation, then a more explicit count is

$$N^{\mathrm{TSP}}_{\mathrm{Classic,edge}}(P) = n \cdot n!. \tag{10.42}$$

A DRCC reduction may group routes by structural features such as local edge blocks, admissible transitions, distance classes, or partial-route fragments.
Let

$$C_P : S_n \to Z(P) \tag{10.43}$$

be such a controlled reduction map.
For a reduced route state $\zeta \in Z(P)$, define

$$\Omega_{\mathrm{TSP}}(\zeta) = \{\sigma \in S_n : C_P(\sigma) = \zeta\}. \tag{10.44}$$

The DRCC count is

$$N^{\mathrm{TSP}}_{\mathrm{DRCC}}(P, \zeta) = N^{\mathrm{TSP}}_{\mathrm{Collapse}}(P) + N^{\mathrm{TSP}}_{\mathrm{Reconstruction}}(P, \zeta). \tag{10.45}$$

The gain in the route-count model is

$$G_{\mathrm{TSP}}(P, \zeta) = \frac{n!}{N^{\mathrm{TSP}}_{\mathrm{Collapse}}(P) + N^{\mathrm{TSP}}_{\mathrm{Reconstruction}}(P, \zeta)}. \tag{10.46}$$

The gain in the edge-accumulation model is

$$G_{\mathrm{TSP,edge}}(P, \zeta) = \frac{n \cdot n!}{N^{\mathrm{TSP}}_{\mathrm{Collapse}}(P) + N^{\mathrm{TSP}}_{\mathrm{Reconstruction}}(P, \zeta)}. \tag{10.47}$$

These expressions are conditional on the selected reduction map and routing reconstruction strategy.

10.9 Case Comparison Table

The following table summarizes the count-level comparison across the case studies.

Case	Classical Count	DRCC Count	Condition for Gain
Graph 3-Coloring	$m3^n$	$\leq 6m + b(G)3^{\omega(G)+1}$	$\omega(G) \ll n$ or slowly growing width.
Full Adder	8	4 structural classes	Output depends only on Hamming weight.
CSP	$\lvert\mathcal{C}\rvert q^n$	$N_{\text{Collapse}} + N_{\text{Reconstruction}}$	Small reconstruction space after constraint-relevant collapse.
SAT	$m2^n$	$N_{\text{Collapse}} + N_{\text{Reconstruction}}$	Clause-vector collapse cheaper than full assignment enumeration.
TSG/TSP	$n!$ or $n \cdot n!$	$N_{\text{Collapse}} + N_{\text{Reconstruction}}$	Route structure admits reusable reduced fragments.

The table shows that DRCC is not a single algorithm with one uniform runtime.

It is a structural runtime framework.

Each problem class requires its own reduction map, reconstruction space, and cost model.

10.10 Transition Point Across Cases

For a size parameter n, the transition condition is

$$N_{\text{Classic}}(n) = N_{\text{DRCC}}(n). \tag{10.48}$$

Equivalently,

$$G(n) = 1. \tag{10.49}$$

If the classical count grows faster than the DRCC count, then the transition point separates two regimes.

For

$$n < n_w,$$

the reduction overhead may dominate.

For

$$n > n_w,$$

the reduced reconstruction model may become favorable.

The transition point is therefore

$$W = (n_w, R_w[s]). \tag{10.50}$$

The value n_w is not universal.

It depends on the problem family, cost model, structural width, reconstruction method, and implementation assumptions.

10.11 Runtime Gap Across Cases

The runtime gap is

$$\Delta_T(n) = T_{\text{Classic}}(n) - T_{\text{DRCC}}(n). \tag{10.51}$$

Since both runtimes use the same reference rate f, the count-level gap is

$$\Delta_N(n) = N_{\text{Classic}}(n) - N_{\text{DRCC}}(n). \tag{10.52}$$

The sign of $\Delta_N(n)$ determines the count-level regime:

$$\Delta_N(n) < 0 \quad \Longrightarrow \quad \text{classical enumeration is smaller under the model,} \tag{10.53}$$

$$\Delta_N(n) = 0 \quad \Longrightarrow \quad \text{transition,} \tag{10.54}$$

and

$$\Delta_N(n) > 0 \quad \Longrightarrow \quad \text{DRCC is smaller under the model.} \tag{10.55}$$

This formulation makes the runtime claim falsifiable at the level of the chosen model.

If the measured or computed DRCC count is not smaller than the classical count, then the instance does not support a DRCC gain under that model.

10.12 Cross-Case Interpretation

Across the case studies, the same pattern appears.

Classical enumeration begins with a large candidate space:

$$X(P).$$

DRCC introduces a controlled reduction map:

$$C_P : X(P) \to Z(P).$$

A reduced state

$$\zeta \in Z(P)$$

defines a reconstruction space:

$$\Omega_P(\zeta).$$

The runtime comparison is then governed by

$$|X(P)| \cdot c_{\text{eval}}(P) \quad \text{versus} \quad N_{\text{Collapse}}(P) + N_{\text{Reconstruction}}(P, \zeta). \tag{10.56}$$

This comparison is the operational core of the condensed runtime manuscript.

It gives a concrete way to ask whether a proposed reduction is useful.

It also gives a concrete way to reject a reduction when collapse and reconstruction overhead exceed the classical baseline.

10.13 No General Complexity Claim

The preceding comparisons do not prove that DRCC solves NP-complete problems in polynomial time.

They do not prove that SAT, graph coloring, CSP, or TSP become easy in general.

They show only that certain finite or structurally restricted instances may admit useful reductions when the reconstruction cost remains controlled.

A valid DRCC runtime claim therefore has the following form:

> For a specified problem instance or instance family, with a specified reduction map, reconstruction space, and cost model, DRCC is favorable when
>
> $$N_{\text{Collapse}} + N_{\text{Reconstruction}} < N_{\text{Classic}}.$$

This is the strongest claim supported by the present runtime framework.

Any stronger statement requires additional proof, implementation, or empirical validation.

10.14 Open Runtime Questions

Several questions remain open.

First, it remains open which broad problem families admit reduction maps whose reconstruction dimensions grow polynomially.

Second, it remains open how often bounded-width reconstruction occurs in real benchmark instances.

Third, it remains open how DRCC-style reductions compare empirically with established methods such as constraint propagation, kernelization, symmetry breaking, dynamic programming, branch-and-bound, and DPLL-type procedures.

Fourth, it remains open whether the transition point W can be predicted from structural parameters alone.

Fifth, it remains open how to design reproducible benchmark protocols for DRCC-style reduction and reconstruction.

These are not weaknesses of the runtime definitions themselves.

They are the natural next questions required for empirical and complexity-theoretic validation.

10.15 Summary

The cross-case analysis uses one common comparison:

$$N_{\text{Classic}}(P) \quad \text{versus} \quad N_{\text{DRCC}}(P, \zeta). \tag{10.57}$$

The DRCC count is

$$N_{\text{DRCC}}(P, \zeta) = N_{\text{Collapse}}(P) + N_{\text{Reconstruction}}(P, \zeta). \tag{10.58}$$

The gain is

$$G(P, \zeta) = \frac{N_{\text{Classic}}(P)}{N_{\text{DRCC}}(P, \zeta)}. \tag{10.59}$$

The transition condition is

$$G(n_w) = 1. \tag{10.60}$$

Equivalently,

$$N_{\mathrm{Classic}}(n_w) = N_{\mathrm{DRCC}}(n_w). \tag{10.61}$$

The main conclusion is conditional:

> DRCC can provide a runtime advantage only when controlled collapse and reconstruction together are cheaper than classical enumeration under the specified cost model.

This condition is explicit, testable, and does not imply a universal speedup theorem.

11 Literature Grounding

11.1 Purpose

This chapter briefly positions DRCC–V2 relative to established approaches in combinatorial search, preprocessing, and constraint-based reasoning.

The purpose is not to claim that DRCC replaces these methods.

The purpose is to clarify that DRCC studies controlled structural collapse and admissible reconstruction in a finite runtime framework.

11.2 Relation to Established Methods

DRCC is related in spirit to several established research directions.

First, kernelization in parameterized complexity studies how instances can be reduced by preprocessing while preserving the relevant decision question. DRCC is not formulated as a kernelization theorem in this manuscript, but the DRCC reduction map $C_P : X(P) \to Z(P)$ has a similar high-level motivation: reduce irrelevant structure while preserving admissible reconstruction.

Second, constraint propagation in constraint satisfaction problems uses local constraints to remove impossible assignments and make implicit constraints explicit. DRCC shares the idea that local structure may control global reconstruction, but DRCC records this through the quantities d_{frag}, d_{rec}, and N_{DRCC}.

Third, SAT solving methods such as DPLL and its successors use branching, propagation, and pruning to search Boolean assignment spaces. DRCC does not replace SAT solvers. It provides a structural language for describing when a clause-induced collapse or reconstruction space may be smaller than full enumeration.

Fourth, symmetry breaking exploits equivalences between search states in order to avoid redundant exploration. DRCC is compatible with this idea, because a controlled reduction class $C_P^{-1}(\zeta)$ may group classically distinct candidates that are equivalent for the chosen reconstruction task.

11.3 Position of DRCC

The contribution of DRCC–V2 is therefore not a new universal solver.

Its contribution is a finite runtime accounting framework based on

$$d_{\text{frag}}(P) = |X(P)|, \tag{11.1}$$

$$d_{\text{rec}}(P, \zeta) = |\Omega_P(\zeta)|, \tag{11.2}$$

and

$$N_{\text{DRCC}}(P, \zeta) = N_{\text{Collapse}}(P) + N_{\text{Reconstruction}}(P, \zeta). \tag{11.3}$$

This makes DRCC comparable to existing methods at the level of instance reduction, reconstruction cost, and transition behavior.

A stronger relation to kernelization, constraint propagation, SAT preprocessing, and symmetry breaking remains future work.

11.4 References

Bibliography

[1] R. G. Downey and M. R. Fellows, *Fundamentals of Parameterized Complexity*, Springer, 2013.

[2] R. Dechter, *Constraint Processing*, Morgan Kaufmann, 2003.

[3] M. Davis, G. Logemann, and D. Loveland, "A Machine Program for Theorem Proving," *Communications of the ACM*, 5(7), 394–397, 1962.

[4] J. Crawford, M. Ginsberg, E. Luks, and A. Roy, "Symmetry-Breaking Predicates for Search Problems," Proceedings of KR/AAAI, 1996.