



ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ

TÔ VĂN KHÁNH · KIỀU VĂN TUYÊN
TRƯƠNG XUÂN HIẾU · VŨ THANH HƯƠNG
ĐÀO XUÂN NGHĨA · NGUYỄN KIM TRUNG ĐỨC

S Á C H C H U Y Ê N K H Ả O

BIỂU DIỄN SAT TỐI ƯU

CHO CÁC BÀI TOÁN TỐI ƯU HÓA TỔ HỢP



ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ

TÔ VĂN KHÁNH · KIỀU VĂN TUYÊN
TRƯƠNG XUÂN HIẾU · VŨ THANH HƯƠNG
ĐÀO XUÂN NGHĨA · NGUYỄN KIM TRUNG ĐỨC

S Á C H C H U Y Ê N K H Ả O

BIỂU DIỄN SAT TỐI ƯU

CHO CÁC BÀI TOÁN TỐI ƯU HÓA TỔ HỢP

Lời nói đầu

Cuốn sách trình bày *SAT encoding*, sau đây gọi là *phép biểu diễn SAT*, như một thành phần trung tâm của thuật toán giải chính xác. Phép biểu diễn chuyển biến và ràng buộc của bài toán gốc thành công thức dạng chuẩn hội CNF. Hai công thức cùng biểu diễn một ràng buộc có thể khác nhau đáng kể về kích thước, khả năng *unit propagation* (lan truyền đơn vị), cấu trúc mệnh đề và mức độ phù hợp với *incremental solving* (giải tăng dần). Vì thế người thiết kế cần đồng thời trả lời bốn câu hỏi: phép biểu diễn có đúng không, trạng thái nào được biểu diễn và chia sẻ, *solver* (bộ giải SAT) nhìn thấy cấu trúc gì, và kết quả tối ưu được chứng nhận thế nào.

Ba phần của sách đi theo một đường liên tục. Phần I tổng hợp nền tảng SAT, MaxSAT, phép chuyển CSP sang SAT, các tiêu chuẩn lan truyền, *cardinality constraint* (ràng buộc đếm) và *pseudo-Boolean constraint* (ràng buộc giả Boolean), cùng phương pháp thực nghiệm. Phần II đặt *shared counter* (Shared Counter), *register* (thanh ghi đếm) thích nghi, *symmetry breaking* (phá vỡ đối xứng) và *channeling* (liên kết biểu diễn) vào các dòng nghiên cứu rộng hơn về ràng buộc SEQUENCE, phân rã ràng buộc toàn cục và cận dưới của phép biểu diễn. Phần III khảo sát bốn miền ứng dụng: lập lịch, đóng gói, các bài toán bandwidth/antibandwidth và gán nhãn/phân bổ tần số.

Các công trình của nhóm tác giả được dùng làm *case study* (nghiên cứu tình huống) để minh họa trọn vẹn quá trình hình thành, chứng minh và đánh giá một phép biểu diễn. Mỗi chương bắt đầu từ hệ phân loại và các kết quả đại diện của cộng đồng, sau đó kết nối chúng với bài toán, kỹ thuật và bằng chứng thực nghiệm của nhóm.

Sách hướng tới học viên cao học, nghiên cứu sinh, giảng viên và người phát triển các mô hình SAT. Người đọc cần kiến thức cơ bản về logic mệnh đề và tối ưu tổ hợp; các khái niệm chuyên sâu được giới thiệu tại nơi sử dụng. Người mới có thể đọc Chương 1–4 trước khi chọn một miền ứng dụng ở Phần III. Người quan tâm ràng buộc đếm nên đọc tiếp Chương 5–6; người nghiên cứu mô hình miền hữu hạn có thể đi từ Chương 2 sang Chương 7 và các chương ứng dụng.

Lời cảm ơn

Nhóm tác giả trân trọng cảm ơn các đồng nghiệp, phản biện và cộng đồng SAT/MaxSAT, *Constraint Programming* (CP, Quy hoạch ràng buộc) và Nghiên cứu vận trù đã công bố những kết quả nền tảng và chia sẻ phần mềm, dữ liệu nghiên cứu mà cuốn sách kế thừa. Những góp ý chuyên môn đã giúp nội dung chính xác và sáng rõ hơn.

Nhóm tác giả đặc biệt cảm ơn các thế hệ sinh viên và cựu sinh viên đã đồng hành trong các seminar, trao đổi học thuật, đọc bản thảo, xây dựng ví dụ và thử nghiệm phần mềm. Những câu hỏi từ góc nhìn người học cùng kinh nghiệm triển khai của các bạn đã giúp nhiều khái niệm trong sách trở nên sáng rõ và gần gũi với thực hành nghiên cứu hơn.

Ký hiệu và quy ước

Bảng 1.: Ký hiệu dùng xuyên suốt cuốn sách.

Ký hiệu	Ý nghĩa
$x, y, z \in \{0, 1\}$	Biến Boolean
$C = \ell_1 \vee \dots \vee \ell_r, F = \bigwedge C$	Mệnh đề và công thức CNF
X, Y	Biến chính và biến phụ
$\text{Enc}(\mathcal{C}; X, Y)$	Phép biểu diễn ràng buộc $\mathcal{C}$
$F \equiv_X \mathcal{C}$	Tương đương theo phép chiếu lên X
$\text{UP}(F, A)$	Bao đóng lan truyền đơn vị dưới phép gán một phần A
AMO , ALO , ExactlyOne	Nhiều nhất một, ít nhất một và đúng một
LB , UB , OPT	Cận dưới, cận trên và giá trị tối ưu được chứng nhận
BKS , best known solution	Nghiệm tốt nhất đã biết

Chỉ số toán học bắt đầu từ 1, trừ khi phần cài đặt ghi rõ khác. Các miền thời gian, tọa độ và nhân là miền nguyên hữu hạn. Mỗi biến phụ được đọc theo *contract* (đặc tả ngữ nghĩa) đã nêu; một literal phụ có ngữ nghĩa chia sẻ khi phép biểu diễn bảo đảm đặc tả tương ứng.

Mục lục

Lời nói đầu	i
Lời cảm ơn	ii
Ký hiệu và quy ước	iii
I. Nền tảng và tiêu chí thiết kế	1
1. SAT và MaxSAT như một nền tảng giải chính xác	2
1.1. Sự giao thoa của ba hướng nghiên cứu	2
1.2. Từ quyết định đến tối ưu	3
1.3. CNF và biến phụ	4
1.3.1. Mệnh đề, literal và mô hình thỏa	4
1.3.2. Tseitin transformation (phép chuyển Tseitin)	4
1.4. Từ DPLL đến CDCL	4
1.5. SAT tăng dần	5
1.6. MaxSAT	6
1.7. Giải mã, ghi chứng minh và chứng nhận	6
1.8. Quy trình triển khai	7
1.9. Bài học thiết kế	7
2. Thế nào là một phép biểu diễn SAT tốt?	9
2.1. Phép biểu diễn là một quan hệ có giao diện	9
2.2. Hệ phân loại theo cách biểu diễn miền	9
2.3. Contract: đặc tả ngữ nghĩa của biến phụ	11
2.4. Đo kích thước	11
2.5. Unit propagation: lan truyền đơn vị	12
2.6. Cận dưới kích thước và biên Pareto (Pareto frontier)	12
2.7. Mức độ phù hợp với bộ giải	13
2.8. Tính mô-đun và biểu diễn lại có cấu trúc	13
2.9. Ma trận phát biểu–bảng chứng	13
2.10. Một quy trình đánh giá phép biểu diễn	14
2.11. Bài học thiết kế	14
3. Bộ công cụ ràng buộc đếm và giả Boolean	15
3.1. Bản đồ phát triển của phép biểu diễn ràng buộc đếm	15
3.2. Các dạng ràng buộc	15
3.3. At-most-one (AMO): ràng buộc nhiều nhất một	16
3.3.1. Biểu diễn từng cặp	16
3.3.2. Biểu diễn tuần tự	16
3.3.3. Biểu diễn tích và biểu diễn chỉ huy	16
3.4. Sequential Counter cho AMK	16
3.5. Totalizer	17
3.6. Mạng đếm	18
3.7. Biểu diễn bằng BDD	18

3.8.	Ràng buộc đúng K và đầu ra chia sẻ	19
3.9.	Pseudo-Boolean có trọng số	19
3.9.1.	Totalizer tổng quát	19
3.9.2.	Khai thác các nhóm AMO trong PB	19
3.10.	Từ lựa chọn thủ công đến lựa chọn theo bài toán thử	20
3.11.	Bản đồ lựa chọn	20
3.12.	Bài học thiết kế	21
4.	Thiết kế thực nghiệm cho phép biểu diễn SAT	22
4.1.	Thực nghiệm là một phép đo có mô hình	22
4.2.	Bắt đầu từ một câu hỏi nhân quả	22
4.3.	Ma trận thiết kế	23
4.4.	Bộ kiểm chuẩn và phân tầng	23
4.5.	Tách tinh chỉnh và đánh giá	23
4.6.	Các thước đo	24
4.6.1.	Kích thước và xây dựng	24
4.6.2.	Quá trình tìm kiếm	24
4.6.3.	Right-censored runtime: thời gian chạy có kiểm duyệt	24
4.6.4.	Biểu đồ Cactus (Cactus plot) và biểu đồ phân tán	25
4.7.	SAT và UNSAT là hai chế độ khác nhau	25
4.8.	Random seed và biến thiên giữa các lần chạy	26
4.9.	So sánh ghép cặp và độ lớn hiệu ứng	26
4.10.	Phân tích theo vùng cấu trúc	27
4.11.	Kiểm đúng trước khi đo nhanh	27
4.12.	Gói tái lập và khả năng truy vết	27
4.13.	Mẫu quy trình thực nghiệm	28
4.14.	Bài học thiết kế	28
II.	Tái sử dụng cấu trúc trong phép biểu diễn SAT	29
5.	Shared Counter cho các cửa sổ chồng lấn	30
5.1.	SEQUENCE trong văn liệu về ràng buộc	30
5.2.	Họ ràng buộc cửa sổ	31
5.3.	Phân hoạch thành khối	31
5.4.	Bộ nối như một giao diện	33
5.5.	Staircase AMO và SCL	33
5.6.	Mở rộng sang Ladder AMK	33
5.6.1.	Quan hệ với tổng tiền tố và ràng buộc chuỗi tổng quát	34
5.7.	Đúng dẫn	34
5.7.1.	Cửa sổ cắt nhiều biên	35
5.8.	Chọn độ rộng khối	35
5.9.	Triển khai	35
5.9.1.	Miền hiệu quả của Shared Counter	36
5.10.	Bài học thiết kế	36
6.	Thanh ghi đếm thích nghi cho ràng buộc đúng K	37
6.1.	Ràng buộc đúng K trong không gian thiết kế	37
6.2.	Từ bảng chữ nhật đến miền trạng thái khả đạt	38

6.3.	Đặc tả ngữ nghĩa và hệ thức truy hồi	38
6.4.	Đóng ràng buộc đúng K	39
6.5.	Cơ chế suy diễn lan truyền hai chiều	40
6.6.	Dùng một thanh ghi cho nhiều giao diện	40
6.7.	Nghiên cứu tình huống ngắn: xếp nhóm chơi golf	40
6.7.1.	Từ thiết kế tổ hợp đến mô hình SAT	40
6.7.2.	Phân tích kết quả	41
6.8.	Cài đặt và kiểm thử	41
6.9.	Miền lựa chọn của NSC	42
6.10.	Bài học thiết kế	42
7.	Đối xứng, biến quan hệ và liên kết biểu diễn	43
7.1.	Hai cấp đối xứng: mô hình và công thức	43
7.2.	Đối xứng như tác động lên nghiệm	44
7.3.	Loại đối xứng ngay từ lựa chọn biến	44
7.4.	Bốn mẫu phá vỡ đối xứng	45
7.4.1.	Cố định một lát chuẩn	45
7.4.2.	Tiền tố sử dụng và thứ tự ưu tiên giá trị	45
7.4.3.	Lex-leader	45
7.4.4.	Neo hình học và phản xạ	46
7.5.	Biến quan hệ cho các phép tuyến	46
7.5.1.	Biến quan hệ và giá trị hỗ trợ	46
7.6.	Order Encoding và các literal so sánh	46
7.7.	Liên kết một chiều hay hai chiều	47
7.7.1.	Lan truyền qua vòng liên kết	48
7.8.	Phép quay và kích thước có điều kiện	48
7.9.	Đồ thị xung đột và tính địa phương	49
7.10.	Kiến trúc kết hợp	49
7.11.	Bài học thiết kế	49
III.	Các họ bài toán tối ưu tổ hợp	51
8.	Lập lịch và cân bằng dây chuyền	52
8.1.	Từ SAT theo mốc thời gian đến lazy clause generation	52
8.2.	Kiến trúc chung của mô hình thời gian	53
8.3.	Lập lịch không ngắt trên các tài nguyên đồng nhất	53
8.3.1.	Mô hình	53
8.3.2.	Từ ràng buộc đếm đến một khối liên tục	54
8.4.	Ràng buộc chuỗi trong lập lịch điều dưỡng	55
8.4.1.	Cửa sổ trượt	55
8.4.2.	Đưa vào mô hình phân ca	55
8.5.	Lập lịch cuộc họp B2B với cân bằng thời gian chờ	56
8.5.1.	Bài toán B2B và mục tiêu khoảng thời gian chờ	56
8.5.2.	Tiền xử lý bảo toàn nghiệm và lọc miền	57
8.5.3.	Biểu diễn thứ tự tiên quyết bằng hậu tố thừa chia sẻ (SPARSE SUFFIX)	57
8.5.4.	Mã hóa ngưỡng thời gian chờ từ nhíp hoạt động	58
8.5.5.	Phương pháp giải và kết quả thực nghiệm	58

8.6.	Peak power: công suất đỉnh trong cân bằng dây chuyền	59
8.6.1.	Mô hình đỉnh công suất	59
8.6.2.	Bốn giao diện tối ưu	59
8.7.	Điều chỉnh lịch tàu bằng MaxSAT-DDD	59
8.7.1.	Tại sao cần rời rạc hóa động	59
8.7.2.	Vòng lặp	60
8.8.	Chọn SAT, SMT, CP hay MIP	61
8.8.1.	Một cây quyết định theo cấu trúc	61
8.9.	Bài học thiết kế	62
9.	Đóng gói và cắt hai chiều	63
9.1.	Bản đồ các phương pháp chính xác cho đóng gói	63
9.2.	Mô hình hình học rời rạc	64
9.3.	Order Encoding cho tọa độ	65
9.3.1.	Ngữ nghĩa	65
9.3.2.	Hướng đặt và liên kết biểu diễn	65
9.4.	Bài toán đóng gói (Packing problem) dải hai chiều	66
9.4.1.	Từ tối ưu chiều cao sang kiểm tra tính khả thi	66
9.4.2.	Ba chế độ giải	66
9.5.	Bài toán đóng gói (Packing problem) vào khung chứa hai chiều	67
9.5.1.	Hai kiến trúc	67
9.6.	Bài toán cắt tấm một kích thước	68
9.6.1.	Mô hình theo số tờ	68
9.6.2.	SAT và MaxSAT	68
9.7.	Giảm miền và đối xứng	68
9.8.	Tách ảnh hưởng của phép biểu diễn, cận và quan hệ trội	69
9.9.	Một thư viện mô-đun cho hình học	69
9.10.	Bài học thiết kế	69
10.	Bandwidth, antibandwidth và tô đa màu	71
10.1.	Bố trí đồ thị, gán nhãn và tô màu	71
10.2.	Phân loại trước khi biểu diễn	72
10.3.	Antibandwidth và cửa sổ staircase	72
10.3.1.	Cận và phương pháp chính xác ngoài SAT	72
10.3.2.	Tính khả thi ở cận k	72
10.3.3.	Thang Sequential Counter	73
10.4.	Antibandwidth tuần hoàn	73
10.4.1.	Khoảng cách vòng	73
10.4.2.	Đối xứng của vòng	74
10.5.	Bài toán No-hole Anti-k-labeling và nén khoảng cấm bằng biến tiền tố	74
10.5.1.	Bài toán gán nhãn phi song ánh và điều kiện toàn ánh	74
10.5.2.	Từ bùng nổ mệnh đề từng cặp đến biểu diễn khoảng bằng biến tiền tố	75
10.5.3.	Phá vỡ đối xứng phản xạ và kết quả thực nghiệm	76
10.6.	Tô màu theo bandwidth	76
10.6.1.	Quan hệ với tô màu đỉnh	76
10.6.2.	Mô hình và sáu họ biểu diễn	77
10.6.3.	Khoảng cấm bằng hai biên	77
10.7.	Tô đa màu theo bandwidth	78
10.7.1.	Mô hình ô màu và giảm miền	78

10.7.2. FE, SBE và OBE	78
10.8. Một trừu tượng chung: giá trị chính xác và khoảng cấm	79
10.9. Quy trình chính xác kết hợp	79
11. Gán nhãn và phân bố tần số	81
11.1. Hai văn liệu nguồn: gán nhãn đồ thị và phân bố tần số	81
11.2. Bổ đề khoảng cấm	82
11.2.1. Quan hệ với Support Encoding	82
11.3. Radio k-labeling	83
11.3.1. Mô hình	83
11.3.2. SAT trực tiếp	83
11.3.3. SAT thủ tự	83
11.3.4. Tối thiểu hóa độ trải theo cách tăng dần	84
11.4. Gán nhãn an toàn k chính xác	84
11.4.1. Mô hình được sử dụng	84
11.4.2. Direct, forward order và reverse order encoding	84
11.5. Phân bố tần số với số giá trị nhỏ nhất	85
11.5.1. Miền rời rạc và hàm mục tiêu	85
11.5.2. Tiền xử lý nhất quán cung	86
11.5.3. DSE và POSE	86
11.5.4. Đếm số tần số phân biệt	86
11.6. So sánh theo cấu trúc	87
11.7. Đối chứng ngoài SAT và cận dưới	87
11.8. Bộ kiểm tra và chứng nhận	88
12. Kết luận và hướng nghiên cứu	89
12.1. Những kết luận bền vững	89
12.2. Tổng hợp các nghiên cứu tình huống	89
12.3. Miền hiệu quả và điểm chuyển thiết kế	90
12.4. Hướng nghiên cứu tiếp theo	91
12.5. Kết luận cuối	91
A. Thư viện PySAT và các Bộ giải Chuyên ngành	92
A.1. Tổng quan Công cụ PySAT (Python Toolkit for SAT Oracles)	92
A.1.1. Giới thiệu và Kiến trúc	92
A.1.2. Minh họa Sử dụng PySAT	92
A.2. Các Bộ giải SAT và MaxSAT	93
A.2.1. Các Bộ giải SAT dựa trên CDCL	93
A.2.2. Các Bộ giải MaxSAT	93
A.3. Các Bộ giải Quy hoạch Ràng buộc (CP Solvers)	93
A.4. Các Bộ giải Quy hoạch Tuyến tính Số nguyên (ILP/MIP Solvers)	94
A.5. Bảng Tổng hợp và Định hướng Lựa chọn Bộ giải	94
Thuật ngữ Anh–Việt	95
Tài liệu tham khảo	98
Chi mục	102

Danh sách hình vẽ

1.1.	Lộ trình lập luận của chương: từ Mô hình miền, qua Biểu diễn CNF, Bộ giải SAT/MaxSAT và Giải mã nghiệm, đến Kiểm tra và cận.	2
1.2.	Tính đơn điệu của bài toán quyết định theo cận B	3
1.3.	Sơ đồ vòng lặp CDCL dưới góc nhìn thiết kế biểu diễn.	5
2.1.	Lộ trình lập luận của chương: từ Tính đúng, qua Kích thước, Lan truyền và Cấu trúc bộ giải, đến Thực nghiệm.	9
2.2.	Hai từ vựng Boolean cho biến $v \in \{1, 2, 3, 4\}$	10
2.3.	Đặc tả của literal phụ quyết định cách các mô-đun khác sử dụng literal đó.	11
3.1.	Lộ trình lập luận của chương: từ AMO/EXO, qua AMK/ALK, Đúng K và Giả Boolean, đến Chọn bộ biểu diễn.	15
3.2.	Lưới trạng thái của Sequential Counter cho $n = 5, k = 2$. Mũi tên liên truyền ngưỡng; mũi tên đứt đưa đầu vào mới vào hệ thức truy hồi.	17
3.3.	Totalizer cân bằng: mỗi nút trả về các ngưỡng của tổng trong cây con.	18
3.4.	Bộ so sánh Boolean là thành phần cơ sở của Sorting Network và mạng đếm.	18
4.1.	Lộ trình lập luận của chương: từ Câu hỏi, qua Ma trận thiết kế, Bộ kiểm chuẩn và Thuốc đo, đến Phân tích.	22
4.2.	Hồ sơ hiệu năng (performance profile): phân đầu cho thấy số bài giải nhanh nhất, phần đuôi cho độ bền so với cấu hình tốt nhất trên từng bài toán thử.	25
4.3.	Biểu đồ Cactus (Cactus plot) (trái) giữ số bài giải được trong hình; biểu đồ phân tán (phải) giữ ghép cặp theo bài toán thử. Các dấu ở biên minh họa lần chạy hết thời hạn của một cấu hình.	26
4.4.	Chuỗi truy vết tối thiểu của một phát biểu thực nghiệm.	27
5.1.	Lộ trình lập luận của chương: từ Chuỗi Boolean, qua Phân hoạch khối, Bộ đếm cục bộ và Bộ nối, đến Họ cửa sổ.	30
5.2.	Ba cửa sổ độ rộng 4 chỉ dịch một vị trí: cửa sổ kề nhau giữ lại ba trong bốn biến. Xây lại một bộ đếm cho từng hàng sẽ lấp phần giao này.	32
5.3.	Một cửa sổ giao biên được phân rã thành hậu tố của khối trái và tiền tố của khối phải; bộ nối chỉ đọc các ngưỡng cần thiết.	32
6.1.	Lộ trình lập luận của chương: từ Dây Boolean, qua Miền trạng thái khả đạt, Truy hồi ngưỡng và Điều kiện biên, đến Đúng K	37
6.2.	Sequential Counter hình chữ nhật và miền trạng thái tam giác. NSC không tạo ba ô bất khả $R_{1,2}, R_{1,3}, R_{2,3}$	38
6.3.	Một lịch khả thi cho bài toán nhỏ 3-2-2. Mỗi ô nhóm là một ràng buộc đúng 2; tuần đầu có thể được cố định làm lát chuẩn.	41
7.1.	Lộ trình lập luận của chương: từ Quyết định gốc, qua Nhóm đối xứng, Đại diện chuẩn và Biến quan hệ, đến Mệnh đề cục bộ.	43
7.2.	Một quỹ đạo gồm bốn cách biểu diễn tương đương. Ràng buộc phá đối xứng hợp lệ giữ ít nhất một nút; phép chuẩn hóa mạnh hơn sẽ cố giữ đúng một nút.	44
7.3.	Bốn lớp hỗ trợ của quan hệ không chồng lấn giữa hai hình chữ nhật. Một literal quan hệ làm chứng cho cả một lớp cặp tọa độ.	47

7.4.	Vòng liên kết trong phép biểu diễn kết hợp. Thiếu một chiều có thể làm đứt một đường lan truyền đơn vị dù công thức vẫn đúng theo phép chiếu.	48
8.1.	Lộ trình lập luận của chương: từ Miền thời gian, qua Biến quyết định, Liên tục và tài nguyên và Hàm mục tiêu, đến Chứng nhận.	52
8.2.	Ràng buộc đếm $\sum_t z_{it} = 3$ không phân biệt một khối liên tục với ba mốc rời; tính không ngắt phải biểu diễn vị trí bắt đầu và hai biên khối.	54
8.3.	Ba cửa sổ ALSC ($w = 4, k = 2$) trên bảy ngày. Mỗi cửa sổ kề nhau chỉ thay một ngày, nên các ngưỡng ở phần giao có thể dùng lại.	55
8.4.	Kỹ thuật biểu diễn trong B2B: (a) Biến hậu tố thừa SPARSE_SUFFIX tại điểm cắt $c = 2$; (b) Xác định ngưỡng thời gian chờ nội bộ từ mốc bắt đầu và mốc kết thúc.	58
8.5.	DDD chỉ thêm mốc tại nơi nghiệm của mô hình hạn chế thiếu thông tin. Lưới không được làm dày đồng loạt trên toàn bộ khung thời gian.	61
9.1.	Lộ trình lập luận của chương: từ Vật và hướng đặt, qua Tọa độ thứ tự, Quan hệ tách và Cận khung chứa, đến Bố trí và chứng nhận.	63
9.2.	Một bố trí nguyên có bốn vật. Các cặp chạm cạnh vẫn hợp lệ; chẳng hạn $x_A + w_A = x_B = 3$, nên biến làm chứng l_{AB} có thể nhận giá trị đúng.	65
9.3.	Hai kiến trúc cho đóng gói vào khung: xếp chồng dùng một trục y toàn cục; không xếp chồng dùng phép gán và tọa độ cục bộ, kèm tiền tố sử dụng.	68
10.1.	Lộ trình lập luận của chương: từ Hoán vị hay tô màu, qua Cửa sổ/thứ tự/khối, Khoảng cách và Tìm cận, đến Gán nhãn và chứng nhận.	71
10.2.	Ở trái, nhãn $(2, 4, 1, 3)$ cho P_4 đạt antibandwidth 2. Ở phải, mỗi ô được tô biểu diễn một cặp nhãn bị cấm trên một cạnh.	73
10.3.	Cửa sổ cấm trên vòng C_8 với tâm 1 và $k = 3$. Hai nhánh 7, 8 và 2, 3 phải được nối khi tuyến tính hóa vòng.	74
10.4.	So sánh cơ chế nén khoảng cấm khi $k = 5, w = 2$: (a) Phép biểu diễn từng cặp liệt kê $O(w)$ mệnh đề; (b) Phép biểu diễn tiền tố PSE loại toàn bộ khoảng cấm bằng một mệnh đề kéo theo duy nhất.	76
11.1.	Lộ trình lập luận của chương: từ Giá trị chính xác, qua Chuỗi thứ tự, Khoảng cấm và Độ trải hoặc số giá trị, đến Nghiệm chứng và cận dưới.	81
11.2.	Một khoảng cấm liên tục được nén thành hai literal biên: $\neg g_{v,4} \vee g_{v,7}$	82
11.3.	Một phép gán nhãn radio 3 khả thi cho P_4 . Hai đỉnh kề nhau cần chênh ít nhất 3, cặp cách hai cạnh cần 2, và hai đầu mút cần 1.	83
11.4.	Ánh xạ khoảng giá trị $[4, 10]$ vào hai biên của miền rời rạc $D_w = \{2, 5, 8, 11\}$	86

Danh sách bảng

1.	Ký hiệu dùng xuyên suốt cuốn sách.	iii
1.1.	Ba tiêu chí hỗ trợ khi đánh giá một phép biểu diễn SAT.	2
1.2.	Vật chứng cần có cho các mức kết luận.	7
2.1.	Các họ biểu diễn miền hữu hạn.	10
2.2.	Ba mức đặc tả của literal ngưỡng.	11
2.3.	Ma trận phát biểu–bằng chứng cho một phép biểu diễn SAT.	13
3.1.	So sánh các bộ biểu diễn AMO cơ bản.	16
3.2.	Nhóm đặc trưng để lựa chọn bộ biểu diễn theo bài toán thử.	20
3.3.	Gợi ý lựa chọn bộ biểu diễn ràng buộc đếm/PB.	20
5.1.	Các cách tái sử dụng trạng thái trong ràng buộc cửa sổ.	30
6.1.	Các hướng biểu diễn ràng buộc đúng K	37
7.1.	Bốn cấp khai thác đối xứng.	43
8.1.	Các mô hình giải chính xác cho bài toán thời gian.	52
8.2.	Bốn giao diện tối ưu cho công suất đỉnh.	59
9.1.	Các biểu diễn chính xác cho đóng gói hai chiều.	63
9.2.	Ba chế độ tối ưu chiều cao bằng SAT/MaxSAT.	66
10.1.	Bốn họ bài toán khoảng cách nhãn.	71
10.2.	Phân biệt ngữ nghĩa của ABP, CABP, No-hole Anti- k -labeling, BCP và BMCP.	72
10.3.	Sáu họ biểu diễn cho tô màu theo bandwidth.	77
11.1.	Ba lớp bài toán gán nhãn và tần số.	81
11.2.	So sánh cấu trúc Radio k , k -Safe và MO-FAP.	87
12.1.	Tổng hợp các nghiên cứu tình huống.	90
A.1.	So sánh đặc tính kỹ thuật của các lớp bộ giải tổ hợp.	94
T.1.	Thuật ngữ Anh–Việt.	95

Danh sách thuật toán

1.1.	Khung DPLL cơ sở	5
5.1.	Sinh AMK cửa sổ chia sẻ khi mỗi cửa sổ cắt nhiều nhất một biên	36
6.1.	Sinh NSC cho ràng buộc đúng K	42
8.1.	Vòng lặp MaxSAT-DDD có kiểm chứng	60

PHẦN I

Nền tảng và tiêu chí thiết kế

1 SAT và MaxSAT như một nền tảng giải chính xác

TÓM TẮT CHƯƠNG

Bộ giải SAT trả lời một câu hỏi quyết định. Để giải tối ưu tổ hợp, ta cần thêm mô hình miền, phép biểu diễn, cơ chế tìm cận và một model checker (bộ kiểm tra độc lập). Bốn lớp này tạo thành một thuật toán hoàn chỉnh.

1.1. Sự giao thoa của ba hướng nghiên cứu

Phép biểu diễn SAT cho tối ưu tổ hợp là sự kết hợp của ba hướng tiếp cận cốt lõi. Thứ nhất, *propositional reasoning* (lập luận mệnh đề) đưa bài toán về CNF để dùng lan truyền đơn vị, phân tích xung đột và học mệnh đề. Thứ hai, *Constraint Programming* (CP, quy hoạch ràng buộc) nhìn mô hình qua biến miền hữu hạn, giá trị hỗ trợ và các mức nhất quán. Thứ ba, *mathematical optimization* (tối ưu toán học) đặt hàm mục tiêu, mô hình nổi lồi, cận dưới và chứng nhận tối ưu ở vị trí trung tâm.

Các công trình về *arc consistency* (nhất quán cung) trong SAT chỉ ra rằng chất lượng phép dịch còn phụ thuộc vào các suy luận miền mà bộ lan truyền tạo được [7, 37]. Một phép biểu diễn có thể đúng về tập nghiệm nhưng làm mất suy luận vốn xuất hiện sớm trong bộ lan truyền của ràng buộc toàn cục. SAT bù lại bằng khả năng học mệnh đề xuyên qua nhiều ràng buộc. Vì vậy sách dùng hai cấp phân tích:

- ở cấp mô-đun, xét *contract* (đặc tả ngữ nghĩa) và suy luận do lan truyền đơn vị;
- ở cấp mô hình, xét tương tác giữa các mô-đun trong CDCL và vòng tối ưu.

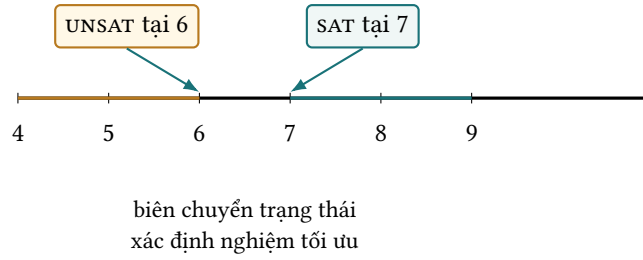
Việc biên dịch CSP miền hữu hạn sang SAT đã phát triển thành một hướng độc lập, với Direct Encoding (biểu diễn trực tiếp), hỗ trợ, thứ tự và lôgarit, cùng nhiều phép phân rã ràng buộc toàn cục [67, 76]. Trong góc nhìn này, “biểu diễn SAT tối ưu” là việc chọn một ngôn ngữ Boolean phù hợp với cấu trúc miền, kiểu suy luận cần giữ và giao diện hàm mục tiêu của bài toán đang xét.

Bảng 1.1.: Ba tiêu chí hỗ trợ khi đánh giá một phép biểu diễn SAT.

Góc nhìn	Đơn vị chính	Bảng chứng mạnh	Rủi ro khi đứng riêng
SAT/CDCL	Literal và mệnh đề	Xung đột, mệnh đề học được, chứng minh	Bỏ qua ngữ nghĩa miền
CP/CSP	Biến, miền, giá trị hỗ trợ	Nhất quán và lọc miền	Không dự báo đầy đủ khả năng học mệnh đề
Tối ưu	Cận và hàm mục tiêu	Khoảng cận, cận dưới, giá trị tối ưu	Mô hình Boolean có thể quá yếu



Hình 1.1.: Lộ trình lập luận của chương: từ Mô hình miền, qua Biểu diễn CNF, Bộ giải SAT/MaxSAT và Giải mã nghiệm, đến Kiểm tra và cận.


 Hình 1.2.: Tính đơn điệu của bài toán quyết định theo cận B .

1.2. Từ quyết định đến tối ưu

Bài toán SAT hỏi liệu một công thức Boolean có phép gán thỏa hay không [14, 25]. Trong ứng dụng tối ưu, ta thường bắt đầu bằng một họ bài toán quyết định phụ thuộc cận B :

$$P(B) : \quad \text{tồn tại nghiệm khả thi } s \text{ sao cho } f(s) \leq B?$$

Với mục tiêu cực tiểu,

$$P(B) = \text{SAT} \implies P(B') = \text{SAT} \quad \forall B' \geq B.$$

Tính đơn điệu cho phép tìm nghiệm tối ưu trên đoạn $[LB, UB]$:

$$\text{OPT} = \min \{B \mid P(B) = \text{SAT}\}.$$

Tìm kiếm tuyến tính từ cận trên thích hợp khi đã có nghiệm tốt và các cận kề nhau chia sẻ nhiều cấu trúc. Tìm kiếm nhị phân giảm số lần gọi bộ giải nhưng nhảy cận xa hơn. Với SAT tăng dần, công thức cơ sở được giữ nguyên; cận được kích hoạt bằng các *assumption* (literal giả định chỉ có hiệu lực trong một lần gọi bộ giải) hoặc bằng các mệnh đề chỉ bổ sung [33].

Trước khi xây dựng thuật toán, cần xác định chính xác phát biểu $P(B)$, chiều đơn điệu và quy tắc cập nhật LB, UB . Bộ giải chỉ trả SAT hoặc UNSAT; ý nghĩa của hai trạng thái đối với cận thuộc về thuật toán bao ngoài.

Giả sử $P(B)$ hỏi liệu có lịch với thời điểm hoàn tất toàn lịch không quá B , và ta biết $LB = 4, UB = 9$. Tìm kiếm nhị phân thử $B = 6$ và nhận UNSAT, nên mọi cận $B \leq 6$ đều bị loại. Thử tiếp $B = 8$ và nhận một lịch khả thi, nên $UB \leftarrow 8$. Cuối cùng $P(7) = \text{SAT}$, cho

$$P(6) = \text{UNSAT}, \quad P(7) = \text{SAT}, \quad \text{OPT} = 7.$$

Mô hình thỏa ở cận 7 là bằng chứng cận trên; chứng minh UNSAT của $P(6)$ là chứng nhận cận dưới. Hai bằng chứng gặp nhau tại giá trị tối ưu 7.

1.3. CNF và biến phụ

1.3.1. Mệnh đề, literal và mô hình thỏa

Một *literal* là biến x hoặc phủ định $\neg x$ của biến đó. Một *clause* (mệnh đề) là phép tuyển các literal; công thức CNF là phép hội của các mệnh đề:

$$F = \bigwedge_{i=1}^m C_i, \quad C_i = \bigvee_{j=1}^{r_i} \ell_{ij}.$$

Một mệnh đề được thỏa khi có ít nhất một literal đúng. Mệnh đề đơn vị có đúng một literal; mệnh đề nhị phân thường biểu diễn phép kéo theo hoặc xung đột cục bộ.

1.3.2. Tseitin transformation (phép chuyển Tseitin)

Biến phụ cho phép thay một biểu thức con bằng một tên ngắn. Với

$$z \leftrightarrow (x \wedge y),$$

CNF tương đương là

$$(\neg z \vee x) \wedge (\neg z \vee y) \wedge (z \vee \neg x \vee \neg y).$$

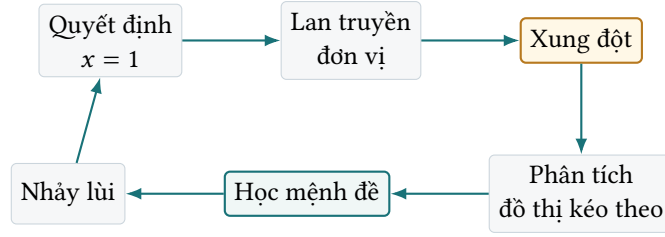
Phép Tseitin giữ kích thước tuyến tính theo mạch Boolean [85]. Ba mệnh đề trên tạo một *contract* (đặc tả ngữ nghĩa): z đúng khi và chỉ khi cả x, y đúng. Nếu mô-đun sau chỉ dùng chiều $z \rightarrow x \wedge y$, có thể bỏ chiều ngược; nhưng khi đó z không còn là tên tương đương của biểu thức.

Biến phụ vừa làm gọn công thức vừa làm lộ trạng thái trung gian cho lan truyền đơn vị và học mệnh đề. Thiết kế phép biểu diễn vì thế là thiết kế một không gian trạng thái cho bộ giải.

Với $z \leftrightarrow (x \wedge y)$, đặt $z = 1$. Hai mệnh đề $(\neg z \vee x)$ và $(\neg z \vee y)$ trở thành mệnh đề đơn vị, nên UP suy $x = y = 1$. Ngược lại, đặt $x = y = 1$; mệnh đề $(z \vee \neg x \vee \neg y)$ suy $z = 1$. Nếu bỏ mệnh đề thứ ba, phép gán $x = y = 1, z = 0$ vẫn thỏa. Khi đó hai mệnh đề còn lại chỉ biểu diễn $z \rightarrow x \wedge y$, không còn tương đương hai chiều. Ví dụ ba biến này là phép kiểm nhanh nên làm trước khi dùng một literal phụ trong mô-đun khác.

1.4. Từ DPLL đến CDCL

DPLL tổ chức phép quyết định SAT như một cây tìm kiếm có lan truyền [14]. Ở mỗi nút, thuật toán chạy lan truyền đơn vị; nếu xuất hiện mệnh đề rỗng thì nhánh hiện tại thất bại, nếu mọi mệnh đề đã thỏa thì thu được mô hình thỏa. Nếu chưa kết luận, chọn một biến x , thử $x = 1$, rồi $x = 0$ nếu nhánh đầu thất bại. Hai nhánh bao phủ mọi phép gán của x , nên lập luận quy nạp theo số biến chưa gán cho tính đầy đủ của thuật toán.



Phép biểu diễn quyết định độ ngắn của đường suy luận và phạm vi áp dụng của mệnh đề học được.

Hình 1.3.: Sơ đồ vòng lặp CDCL dưới góc nhìn thiết kế biểu diễn.

DPLL(F, α)

1. Lắp lan truyền đơn vị trên F dưới phép gán α .
2. Nếu có mệnh đề rỗng, trả UNSAT; nếu mọi mệnh đề đã thỏa, trả SAT và α .
3. Chọn một biến chưa gán x . Gọi $\mathbf{DPLL}(F, \alpha \cup \{x = 1\})$; nếu nhận SAT, trả mô hình đó.
4. Trả kết quả của $\mathbf{DPLL}(F, \alpha \cup \{x = 0\})$.

Thuật toán 1.1: Khung DPLL cơ sở

DPLL cơ sở quay lui theo thứ tự quyết định và không lưu một mô tả tổng quát của nhánh đã thất bại. CDCL giữ bộ khung quyết định–lan truyền–xung đột nhưng ghi lý do của mỗi literal trên *trail* (vết gán theo thứ tự), phân tích đồ thị kéo theo, học một mệnh đề và thực hiện *backjump* (nhảy lùi đến mức quyết định thích hợp). *Restart* (khởi động lại) làm mới phần quyết định nhưng giữ các mệnh đề đã học. Bước chuyển này giải thích vì sao phép biểu diễn không chỉ ảnh hưởng đến số lượng nút tìm kiếm mà còn quyết định các cạnh lý do mà phân tích xung đột có thể sử dụng.

Các bộ giải SAT hiện đại chủ yếu dựa trên *conflict-driven clause learning* (học mệnh đề theo xung đột, CDCL) [32, 58]. Một vòng CDCL gồm:

1. chọn một literal quyết định;
2. chạy lan truyền đơn vị;
3. khi có xung đột, phân tích đồ thị kéo theo;
4. học mệnh đề và nhảy lùi;
5. tiếp tục cho đến mô hình thỏa hoặc chứng minh UNSAT.

Phép biểu diễn ảnh hưởng trực tiếp ba lớp của vòng này. Mệnh đề nhị phân và tam phân tạo đường lan truyền ngắn. Biến phụ quyết định hình dạng đồ thị kéo theo. Cấu trúc mô-đun và tính địa phương ảnh hưởng chất lượng mệnh đề học được; chỉ số Chỉ số LBD là thước đo cốt lõi để quản lý và dọn dẹp mệnh đề xung đột [6].

Vì vậy hai phép biểu diễn có cùng số mệnh đề vẫn có thể khác về thời gian chạy. Phép biểu diễn lớn hơn đôi khi chạy nhanh hơn nhờ các trạng thái trung gian giúp bộ giải học mâu thuẫn có tính khái quát.

1.5. SAT tăng dần

Một bộ giải tăng dần giữ các mệnh đề đã học qua nhiều lời gọi. Có hai cơ chế chính.

Bổ sung mệnh đề đơn điệu. Công thức cơ sở F_0 được mở rộng

$$F_0 \subseteq F_1 \subseteq \dots \subseteq F_t.$$

Đây là dạng tự nhiên cho tối ưu từ trên xuống: sau một mô hình có giá trị β , thêm mệnh đề buộc hàm mục tiêu tốt hơn.

Giả định. Một tập literal A_i chỉ có hiệu lực trong lần gọi t . Các mệnh đề cơ sở vẫn giữ nguyên; giả định cũ được thu hồi khi lời gọi kết thúc. Literal kích hoạt a_B cho phép viết

$$a_B \rightarrow \text{Bound}(B)$$

và bật cận bằng giả định a_B .

Mệnh đề học hợp lệ khi được bộ giải suy ra theo đúng quy trình giả định. Công thức cơ sở, literal kích hoạt và thứ tự gọi là một giao diện duy nhất.

1.6. MaxSAT

Partial MaxSAT (MaxSAT bộ phận) chia công thức thành mệnh đề cứng H và mệnh đề mềm S . Mệnh đề cứng phải được thỏa; hàm mục tiêu tối thiểu số hoặc tổng trọng số của các mệnh đề mềm bị vi phạm:

$$\min_{\alpha \models H} \sum_{C \in S} w_C[\alpha \models C].$$

MaxSAT phù hợp khi mục tiêu có dạng tổng các quyết định Boolean, chẳng hạn số thùng được dùng, số ràng buộc mềm bị vi phạm hoặc tổng mức phạt [4, 55].

Một cách giải MaxSAT là chuyển hàm mục tiêu thành chuỗi bài toán SAT với cận chi phí [29]. Vì thế khác biệt giữa SAT tăng dần và MaxSAT thường nằm ở *giao diện hàm mục tiêu*, còn phần biểu diễn tính khả thi có thể được chia sẻ.

Văn liệu MaxSAT hiện đại thường phân biệt ba kiến trúc. *Linear/binary search* (tìm kiếm tuyến tính/nhị phân) đặt cận lên tổng các literal nổi lóng. *Core-guided* (dẫn dắt bằng lõi) dùng một lõi bất khả thỏa để nhận diện nhóm mệnh đề mềm không thể đồng thời thỏa, rồi nói chúng có kiểm soát. *Hitting-set* (tập đánh trúng) phân tách giai đoạn tìm lõi xung đột khỏi giai đoạn chọn các mệnh đề mềm cần vi phạm. Các bộ giải dạng mô-đun như Open-WBO và RC2 đặt phép biểu diễn đếm/PB ở trung tâm thuật toán: mô-đun này được dựng, mở rộng hoặc thay cận nhiều lần trong quá trình giải MaxSAT [44, 60].

Ràng buộc đếm tăng dần giúp tránh xây lại hàm mục tiêu sau mỗi lõi hoặc cận trên [59]. Vì thế khi chọn cây tổng, bộ đếm hay mạng cho hàm mục tiêu, cần hỏi thêm: cấu trúc có hỗ trợ thêm đầu vào không; đầu ra có đọc được ở nhiều cận không; và các mệnh đề đã học có còn hợp lệ sau lần cập nhật không.

Ví dụ 1.1 (Tối thiểu số tài nguyên). Giả sử $u_j = 1$ khi tài nguyên j được sử dụng. Các ràng buộc gán bảo đảm mỗi tác vụ chọn đúng một tài nguyên và

$$x_{ij} \rightarrow u_j.$$

Cách SAT theo cận thêm

$$\sum_j u_j \leq B.$$

Cách MaxSAT giữ các mệnh đề cứng và đặt các mệnh đề mềm $\neg u_j$ có trọng số 1. Hai mô hình chia sẻ các mô-đun gán và xung đột.

1.7. Giải mã, ghi chứng minh và chứng nhận

Mô hình thỏa của bộ giải chứa cả biến chính và biến phụ. Bộ giải mã chỉ đọc biến chính để tạo nghiệm miền. Bộ kiểm tra sau đó xác minh:

1. mọi giá trị nằm trong miền;
2. mọi ràng buộc của bài toán gốc;
3. hàm mục tiêu được tính lại từ nghiệm;
4. cận và trạng thái bộ giải nhất quán.

Bộ kiểm tra được cài đặt độc lập với bộ biểu diễn để hai thành phần tạo thành một phép kiểm chéo.

Với giá trị tối ưu B^* , bằng chứng cận trên là một nghiệm có chi phí B^* . Chứng cứ cận dưới có thể là chứng minh UNSAT của $P(B^* - 1)$ hoặc một cận toán học bằng B^* . Mô hình SAT chứng minh sự tồn tại; chứng minh UNSAT loại mọi nghiệm ở cận tốt hơn.

Trong thực hành, bộ giải có thể xuất chứng minh theo một định dạng mệnh đề như DRAT; bộ kiểm tra độc lập kiểm lại chuỗi phép thêm/xóa mệnh đề [90]. Điều này tạo một chuỗi tin cậy ngắn hơn so với việc đặt niềm tin hoàn toàn vào bộ giải, nhưng chỉ khi công thức được chứng minh đúng là công thức của cận cần xét. Với giả định, ta hiện thực hóa (reification) các giả định cuối thành mệnh đề đơn vị, ghi chính xác mã kiểm tra của CNF và kiểm chứng minh trên *artifact* (gói tái lập) đó.

Bảng 1.2.: Vật chứng cần có cho các mức kết luận.

Kết luận	Vật chứng	Bộ kiểm tra độc lập
Khả thi tại B	Mô hình/ <i>witness solution</i> (nghiệm làm chứng)	Kiểm ràng buộc và chi phí miền
Vô nghiệm tại B	Chứng minh mệnh đề	Bộ kiểm chứng trên đúng CNF
Tối ưu B^*	Nghiệm tại B^* và chứng nhận cận dưới	Bộ kiểm nghiệm và bộ kiểm chứng cận

1.8. Quy trình triển khai

1. Đọc bài toán đầu vào và chuẩn hóa miền.
2. Tính cận dưới và cận trên.
3. Sinh biến, ánh xạ và CNF cơ sở.
4. Chạy SAT/MaxSAT theo một quy trình tìm cận.
5. Giải mã mô hình sau mỗi lần SAT.
6. Kiểm nghiệm và hàm mục tiêu bằng bộ kiểm tra độc lập.
7. Lưu nghiệm cùng chứng nhận cận dưới ở cận cuối.

PySAT cung cấp môi trường thuận tiện để thử nghiệm nhiều bộ giải và phép biểu diễn [43]; ánh xạ biến và bộ kiểm tra vẫn thuộc về ứng dụng. Một dây chuyền xử lý tốt có thể thay bộ giải mà giữ nguyên ngữ nghĩa.

1.9. Bài học thiết kế

1. SAT là tầng quyết định; tối ưu cần quy trình tìm cận hoặc MaxSAT.
2. Biến phụ phải có đặc tả ngữ nghĩa, không chỉ có một số DIMACS.

3. Giải tăng dần hiệu quả khi công thức cơ sở ổn định và cập thay qua một giao diện rõ.
4. Bộ giải mã và bộ kiểm tra là một phần của thuật toán chính xác.
5. Chứng nhận tối ưu cần cả nghiệm và chứng nhận cận dưới.

Câu hỏi nghiên cứu

1. Viết $P(B)$ và quy tắc cập nhật cận cho một bài toán cực đại.
2. Bỏ từng mệnh đề trong phép biểu diễn $z \leftrightarrow (x \wedge y)$ và tìm phép gán làm mất một chiều tương đương.
3. Mô tả hai cách biểu diễn “tối thiểu số nhóm được dùng”: SAT theo cận và *Partial MaxSAT* (MaxSAT bộ phận).

2 Thế nào là một phép biểu diễn SAT tốt?

TÓM TẮT CHƯƠNG

Chất lượng phép biểu diễn được đánh giá trên nhiều trục. Đúng đắn là điều kiện bắt buộc; các trục còn lại gồm kích thước, sức lan truyền, tính địa phương, chi phí tích hợp và bằng chứng thực nghiệm.

2.1. Phép biểu diễn là một quan hệ có giao diện

Cho ràng buộc $\mathcal{C}$ trên tập biến chính X , tập biến phụ Y và

$$F = \text{Enc}(\mathcal{C}; X, Y).$$

Tiêu chuẩn cơ bản là tương đương theo phép chiếu:

$$F \equiv_X \mathcal{C} \iff \forall \alpha_X : \alpha_X \models \mathcal{C} \iff \exists \beta_Y : \alpha_X \cup \beta_Y \models F.$$

Hai chiều có tên riêng:

- **soundness (tính đúng)**: mọi mô hình thỏa của F chiếu thành nghiệm của $\mathcal{C}$;
- **completeness (tính đầy đủ)**: mọi nghiệm của $\mathcal{C}$ mở rộng được thành mô hình thỏa của F .

Tính đúng thường được chứng minh bằng cách lấy một mô hình CNF và giải mã. Tính đầy đủ thường được chứng minh bằng cách xây biến phụ từ một nghiệm miền. Với bộ đếm, cách xây tự nhiên là đặt *register* (thanh ghi ngưỡng, tức biến phụ lưu một mức đếm) đúng theo tổng tiền tố thật.

Mệnh đề 2.1 (Mẫu chứng minh bằng phép chiếu). *Giả sử bộ giải mã D ánh mọi mô hình thỏa của F thành nghiệm của $\mathcal{C}$, và với mọi nghiệm s của $\mathcal{C}$ tồn tại một phép gán biến phụ $\beta(s)$ làm F đúng. Khi đó $F \equiv_X \mathcal{C}$.*

Chứng minh. Bộ giải mã cho tính đúng. Hàm β cho tính đầy đủ. Hai chiều đúng với mọi phép gán trên X , nên tương đương theo phép chiếu. $\square$

2.2. Hệ phân loại theo cách biểu diễn miền

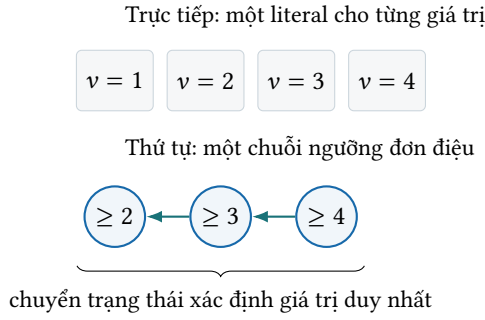
Trước khi so hai bộ biểu diễn, cần xác định chúng có cùng “tù vạng Boolean” hay không. Với biến miền hữu hạn $v \in D = \{d_1, \dots, d_m\}$, bốn họ thường gặp là:

Direct Encoding (biểu diễn trực tiếp)

$x_{v,a}$ mang nghĩa $v = a$. Ràng buộc miền cần đúng-một; xung đột giữa hai giá trị được viết trực tiếp.



Hình 2.1.: Lộ trình lập luận của chương: từ Tính đúng, qua Kích thước, Lan truyền và Cấu trúc bộ giải, đến Thực nghiệm.



Hình 2.2.: Hai từ vựng Boolean cho biến $v \in \{1, 2, 3, 4\}$.

Support Encoding (biểu diễn hỗ trợ)

Mỗi lựa chọn $v = a$ kéo theo một phép tuyển các giá trị hỗ trợ của biến bên kia. Cách này thường phản ánh nhất quán cung tốt nhưng các mệnh đề hỗ trợ có thể dài.

Order Encoding (biểu diễn thứ tự)

$o_{v,t}$ mang nghĩa $v \geq t$ hoặc $v \leq t$. Bất đẳng thức, khoảng và quan hệ trước-sau trở thành phép kéo theo giữa các ngưỡng [76].

Log/binary encoding (biểu diễn lôgarit/nhị phân)

Giá trị dùng $\lceil \log_2 m \rceil$ bit. Số biến nhỏ, nhưng một giá trị hoặc khoảng thường không còn là một literal đơn; lan truyền miền có thể đi qua mạch sâu hơn.

Biểu diễn lai giữ hai góc nhìn, chẳng hạn biến giá trị để chọn một giá trị và biến thứ tự để biểu diễn một khoảng. Chi phí liên kết hai biểu diễn chỉ đáng bỏ ra khi góc nhìn thứ hai rút ngắn một họ ràng buộc lớn hoặc tạo suy luận mới. Đây là kiến trúc được dùng lại trong Chương 7, 9–11.

Bảng 2.1.: Các họ biểu diễn miền hữu hạn.

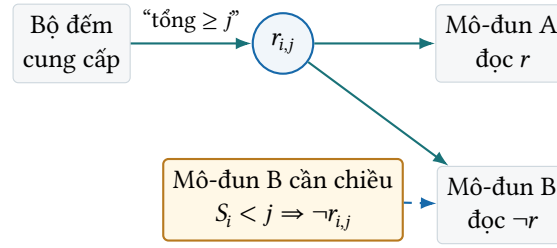
Họ	Literal tự nhiên	Phù hợp	Điểm phải kiểm tra
Trực tiếp	$v = a$	Bảng cấm, miền nhỏ	Đúng-một và nhiều cặp xung đột
Hỗ trợ	" $v = a$ có hỗ trợ"	Ràng buộc nhị phân dạng bảng	Độ dài mệnh đề hỗ trợ
Thứ tự	$v \geq t$	Cận, khoảng, hiệu	Miền rộng và liên kết với biến giá trị
Lôgarit	Bit của giá trị	Miền lớn, mạch số học	Giá trị ngoài miền, sức lan truyền
Lai	Giá trị + ngưỡng/quan hệ	Nhiều kiểu ràng buộc	Chi phí và chiều liên kết

Cho $u, v \in \{0, \dots, 4\}$ và $u + 2 \leq v$. Direct Encoding cấm từng cặp (a, b) với $a + 2 > b$:

$$\neg x_{u,a} \vee \neg x_{v,b}.$$

Order Encoding dùng $o_{z,t} \leftrightarrow (z \geq t)$ và các mệnh đề

$$\neg o_{u,t} \vee o_{v,t+2}, \quad t = 1, 2.$$



Hình 2.3.: Đặc tả của literal phụ quyết định cách các mô-đun khác sử dụng literal đó.

Nếu một ràng buộc khác suy $u \geq 2$, UP đặt $o_{u,2} = 1$, rồi suy $o_{v,4} = 1$, tức $v \geq 4$. Một mệnh đề thứ tự đã thay toàn bộ các cặp trực tiếp có $u \geq 2, v \leq 3$. Đổi lại, giá trị chính xác của v chỉ xuất hiện qua điểm chuyển của chuỗi.

2.3. Contract: đặc tả ngữ nghĩa của biến phụ

Một thanh ghi $r_{i,j}$ có thể được định nghĩa theo ba mức:

Bảng 2.2.: Ba mức đặc tả của literal ngưỡng.

Đặc tả	Công thức	Cách sử dụng
Điều kiện đủ	$r_{i,j} \rightarrow S_i \geq j$	$r = 1$ chứng minh đạt ngưỡng
Điều kiện cần	$S_i \geq j \rightarrow r_{i,j}$	$r = 0$ chứng minh chưa đạt ngưỡng
Tương đương	$r_{i,j} \leftrightarrow (S_i \geq j)$	Cả hai cực tính đều có ngữ nghĩa

Một *connector* (bộ nối giữa các mô-đun) dùng $\neg r_{i,j}$ như mệnh đề “tổng dưới j ” phải có chiều cần. Nếu bộ đếm chỉ cung cấp chiều đủ, bộ nối đó không đúng dù khi đúng riêng hai công thức vẫn có thể *equisatisfiable* (đồng khả thỏa, tức cùng có nghiệm hoặc cùng vô nghiệm).

Trước khi nối hai mô-đun, đối chiếu cực tính mà bên sử dụng cần với chiều đặc tả mà bên cung cấp bảo đảm. Nếu hai phía cần đọc cùng một biến theo hai cách, phải có mệnh đề liên kết tương ứng.

2.4. Đo kích thước

Bốn đại lượng nền tảng là:

$$n_v = |Y|, \quad n_c = |F|, \quad n_\ell = \sum_{C \in F} |C|, \quad w_{\max} = \max_{C \in F} |C|.$$

Ngoài độ lớn tiệm cận, cần xem phân bố mệnh đề nhị phân/tam phân, số lần xuất hiện của biến và bộ nhớ của bộ sinh. Với kỹ thuật hai literal được theo dõi, tổng số literal thường gần chi phí bộ nhớ hơn số mệnh đề.

Một công thức đóng về n_v, n_c phải chỉ rõ:

- miền tham số, đặc biệt $k = 0, 1, n$;
- thanh ghi biên nào được bỏ;

- khối cuối có thể ngắn;
- mệnh đề logic nào cần Tseitin hóa thêm.

Bộ sinh nên tự xuất thống kê và so với công thức lý thuyết trên các bài toán nhỏ. Đây là một kiểm thử đơn vị cho phép đếm, đồng thời phát hiện mệnh đề bị lặp hoặc thanh ghi vô dụng.

2.5. Unit propagation: lan truyền đơn vị

Cho phép gán một phần A . *Unit propagation* (lan truyền đơn vị, UP) lặp lại quá trình thỏa mãn mệnh đề đơn vị cho đến điểm cố định hoặc xung đột. Có ba mức phát biểu thường gặp:

UP phát hiện vi phạm

Nếu A không mở rộng được thành nghiệm của $\mathcal{C}$, UP trên $F \wedge A$ tạo xung đột.

Unit-refutation completeness (URC)

Mọi phép gán làm F không thỏa được đều bị UP chứng minh bất khả thỏa trên toàn bộ biến; sách gọi tính chất này là *đầy đủ bác bỏ bằng lan truyền đơn vị*.

Propagation completeness (PC)

Mọi literal bị $F \wedge A$ kéo theo trong phạm vi quan sát đều được UP suy ra, hoặc UP chứng minh bất khả thỏa; đây là *đầy đủ lan truyền*.

Trong một phép biểu diễn miền hữu hạn, *generalized arc consistency* (nhất quán cung tổng quát, GAC) loại mọi giá trị không còn giá trị hỗ trợ. Phát biểu GAC luôn đi cùng cách Boolean hóa miền và tập literal đại diện cho giá trị.

Kết quả của Gent và Bacchus cung cấp một cầu nối chính xác: nếu literal Boolean đại diện trực tiếp cho khả năng duy trì giá trị trong miền, có thể so sánh bao đóng của lan truyền đơn vị với bao đóng của một bộ lan truyền GAC [7, 37]. Khi thêm liên kết biểu diễn, đổi từ mã hóa trực tiếp sang thứ tự hoặc chỉ giữ một chiều Tseitin, tập literal quan sát thay đổi; phát biểu về GAC vì thế cũng được xác định lại theo tập này.

Ví dụ 2.2 (AMO từng cặp). Với

$$\text{AMO}(x_1, \dots, x_n) = \bigwedge_{i < j} (\neg x_i \vee \neg x_j),$$

nếu $x_i = 1$, UP lập tức suy $\neg x_j$ cho mọi $j \neq i$. Nếu hai biến đúng, UP tạo xung đột. AMO từng cặp vì thế có lan truyền rất trực tiếp trên biến chính, đổi lại $O(n^2)$ mệnh đề.

2.6. Cận dưới kích thước và biên Pareto (Pareto frontier)

Yêu cầu lan truyền đặt ra cận dưới cho kích thước phép biểu diễn. Với AMO, các cận dưới đã biết cho thấy một phép biểu diễn đầy đủ lan truyền cần số mệnh đề tuyến tính với hệ số bị chặn, kể cả khi dùng biến phụ [54]. Kết quả này dẫn đến ba nguyên tắc:

1. xác định rõ lớp biểu diễn khi phát biểu về kích thước, biến phụ và sức lan truyền;
2. một cải tiến vài mệnh đề có thể đáng kể nếu mức giảm đó tiến gần cận dưới lý thuyết;
3. so sánh kích thước phải giữ cùng tiêu chuẩn lan truyền và cùng ngữ nghĩa của đầu ra.

Các kết quả về phép phân rã ALLDIFFERENT và ràng buộc đếm toàn cục còn liên hệ yêu cầu lan truyền với cận biểu diễn của mạch đơn điệu [11]. Khi một ràng buộc toàn cục có bộ lan truyền đạt mức

nhất quán mạnh hơn phép biểu diễn đa thức đang xét, phép so sánh tập trung vào phần suy luận được chuyển sang học mệnh đề và chi phí để đạt phần suy luận đó.

Một bộ biểu diễn được định vị trên bốn trục: kích thước, sức lan truyền, chi phí xây dựng và khả năng tăng dần. Điểm Pareto là điểm mà cải thiện thêm một trục sẽ đánh đổi ít nhất một trục khác. *Benchmark* (bộ kiểm chuẩn) đo vị trí thực nghiệm của từng thiết kế trên biên Pareto (Pareto frontier) này.

2.7. Mức độ phù hợp với bộ giải

Hai phép biểu diễn có cùng tính đúng và kích thước vẫn có thể khác về:

- **tính địa phương:** một ràng buộc cục bộ dùng các biến gần nhau hay đi qua một cấu trúc toàn cục;
- **độ sâu kéo theo:** số bước từ quyết định đến xung đột;
- **khả năng học:** xung đột có tạo mệnh đề tái sử dụng được không;
- **đối xứng:** trạng thái phụ có tạo nhiều mô hình tương đương;
- **khả năng tăng dần:** đầu ra có đọc được ở nhiều cận không.

Không có một công thức giải tích đơn giản cho các thuộc tính này. Ta dùng đại lượng thay thế như số mệnh đề nhị phân, đồ thị kéo theo trên bài toán nhỏ, hoặc số lần lan truyền/xung đột; kết luận cuối cùng cần bộ kiểm chuẩn có kiểm soát.

2.8. Tính mô-đun và biểu diễn lại có cấu trúc

Một mô-đun biểu diễn tốt có bốn thành phần:

variables + clauses + decoder + contract.

Khi nhiều ràng buộc chia sẻ biểu thức con, biểu diễn lại có cấu trúc tạo biến phụ cho phần chung thay vì Tseitin hóa từng công thức riêng [40]. Lợi ích xuất hiện khi mô-đun thực sự được dùng chung trong CNF và bên sử dụng đọc đúng đặc tả.

Ví dụ, nhiều cửa sổ cùng đọc ngưỡng của một tiền tố. Một bộ đếm tiền tố dùng chung giảm lặp; bộ nối chỉ ghép các ngưỡng ở hai biên. Đây là chủ đề trung tâm của Chương 5.

2.9. Ma trận phát biểu–bằng chứng

Bảng 2.3.: Ma trận phát biểu–bằng chứng cho một phép biểu diễn SAT.

Phát biểu	Bảng chứng lý thuyết	Bảng chứng thực nghiệm
Biểu diễn đúng	Chứng minh hai chiều	Kiểm thử vết cạn/vi sai trên miền nhỏ
Biểu diễn nhỏ hơn	Công thức đếm	Kích thước CNF thực tế
Lan truyền mạnh hơn	Bổ đề UP/PC/GAC	Bộ đếm lan truyền/xung đột
Nhanh hơn	Phân tích cơ chế	Bộ kiểm chuẩn, nhiều <i>random seed</i> (hạt giống ngẫu nhiên), <i>performance profile</i> (hồ sơ hiệu năng)
Chứng nhận tối ưu	Định lý về cạn	Nghiệm và chứng minh ở hai cận kề

Ma trận đưa mỗi ý tưởng vào đúng loại đóng góp: định lý, thiết kế cấu trúc hay quan sát thực nghiệm.

2.10. Một quy trình đánh giá phép biểu diễn

- Viết rõ ngữ nghĩa của biến chính và biến phụ.
- Chứng minh tính đúng và tính đầy đủ theo phép chiếu.
- Đếm biến, mệnh đề, literal và độ rộng mệnh đề theo đúng biến thể.
- Phân tích một số suy luận cốt lõi của lan truyền đơn vị (UP).
- Xây dựng *oracle* (bộ đối chiếu nhỏ trên miền đã biết) để so sánh tập nghiệm sau phép chiếu.
- Đánh giá trên bộ kiểm chuẩn theo quy trình của Chương 4.
- Kiểm tra khả năng kết hợp mô-đun và giải tăng dần.

2.11. Bài học thiết kế

- Tính đúng là quan hệ theo phép chiếu, không phải số mô hình của biến phụ.
- Đặc tả ngữ nghĩa quyết định biến phụ có thể được tái sử dụng thế nào.
- Đếm literal và cấu trúc mệnh đề quan trọng ngang với số mệnh đề.
- Phát biểu về sức lan truyền phải nêu rõ phạm vi literal được quan sát.
- Thời gian chạy là kết quả của tương tác giữa phép biểu diễn, bộ giải và bài toán thử.

Câu hỏi nghiên cứu

- Cho một phép biểu diễn AMO có biến phụ, viết bộ giải mã và chứng minh theo phép chiếu.
- Thiết kế một phép thử phân biệt đặc tả điều kiện đủ với đặc tả tương đương của thanh ghi ngưỡng.
- Chọn ba thước đo để giải thích vì sao một phép biểu diễn lớn hơn lại chạy nhanh hơn trên các bài toán thử UNSAT.

3 Bộ công cụ ràng buộc đếm và giả Boolean

TÓM TẮT CHƯƠNG

Cardinality constraint (ràng buộc đếm) giới hạn số literal đúng trong một tập; pseudo-Boolean constraint (ràng buộc giả Boolean) mở rộng mô hình này bằng các trọng số nguyên. Hai lớp này xuất hiện trong ràng buộc gán, súc chứa, chuỗi, đối xứng và hàm mục tiêu. Không có một bộ biểu diễn tốt nhất cho mọi miền; cần chọn theo n , ngưỡng k , các đầu ra cần đọc và cách bộ giải thay cận.

3.1. Bản đồ phát triển của phép biểu diễn ràng buộc đếm

Các phép biểu diễn ràng buộc đếm không phát triển theo một chuỗi trong đó bộ mã hóa mới thay thế hoàn toàn bộ biểu diễn cũ. Mỗi họ giải quyết một kiểu đánh đổi:

1. *pairwise encoding* (biểu diễn từng cặp) và các phép phân rã AMO ưu tiên phát hiện xung đột trực tiếp;
2. *Sequential Counter* (bộ đếm tuần tự) hiện thực hóa (reification) tổng trên từng tiền tố;
3. *Totalizer* (cây tổng) hiện thực hóa (reification) tổng theo cấu trúc cây;
4. *Sorting Network / Selection Network* (mạng sắp xếp / lựa chọn) ưu tiên cấu trúc đều và độ sâu nhỏ;
5. *binary decision diagram* (biểu đồ quyết định nhị phân, BDD) hợp nhất các trạng thái tổng tương đương;
6. *Totalizer* tổng quát và các biến thể PB xử lý trọng số;
7. danh mục phương pháp hoặc bộ chọn đã học chọn phép biểu diễn theo đặc trưng của bài toán thử.

Các khảo cứu về ràng buộc đếm nhấn mạnh rằng phải tách ba thuộc tính: kích thước tiệm cận, mức nhất quán do lan truyền đơn vị duy trì và hiệu năng trong một bộ giải cụ thể [8]. Một Sequential Counter $O(nk)$ có thể nhỏ khi k bé; một mạng đếm có thể lớn hơn nhưng có độ sâu tốt; Totalizer lại thuận lợi khi cận thay đổi. Không có quan hệ thứ tự tuyến tính giữa ba lựa chọn đó.

3.2. Các dạng ràng buộc

Với $X = \{x_1, \dots, x_n\}$, ba tên tiếng Anh lần lượt là *at-most-k* (nhiều nhất k , AMK), *at-least-k* (ít nhất k , ALK) và *exactly-k* (đúng k , EXK):

$$\text{AMK}(X, k) : \sum_{i=1}^n x_i \leq k,$$



Hình 3.1.: Lộ trình lập luận của chương: từ AMO/EXO, qua AMK/ALK, Đúng K và Giả Boolean, đến Chọn bộ biểu diễn.

$$\begin{aligned} \text{ALK}(X, k) : \quad & \sum_{i=1}^n x_i \geq k, \\ \text{EXK}(X, k) : \quad & \sum_{i=1}^n x_i = k. \end{aligned}$$

At-most-one (nhiều nhất một, AMO) là AMK với $k = 1$; *at-least-one* (ít nhất một, ALO) là một mệnh đề dài $x_1 \vee \dots \vee x_n$; *exactly-one* (đúng một, EXO) là AMO kết hợp ALO.

Đối ngẫu hữu ích:

$$\sum_i x_i \geq k \iff \sum_i (1 - x_i) \leq n - k.$$

Tuy nhiên, đối ngẫu có thể đổi ngưỡng từ nhỏ sang lớn. Vì vậy, lựa chọn giữa Direct Encoding ALK và biểu diễn AMK trên các literal phụ định phụ thuộc vào k và những đầu ra đã có.

3.3. At-most-one (AMO): ràng buộc nhiều nhất một

3.3.1. Biểu diễn từng cặp

Biểu diễn từng cặp sinh

$$\neg x_i \vee \neg x_j \quad \forall i < j.$$

Phương pháp này không dùng biến phụ, có $\binom{n}{2}$ mệnh đề nhị phân và lan truyền trực tiếp. Đây là lựa chọn tốt khi miền nhỏ.

3.3.2. Biểu diễn tuần tự

Đặt $s_i = 1$ khi ít nhất một trong $x_1, \dots, x_i$ đúng. Một biến thể biểu diễn AMO tuần tự dùng:

$$\neg x_i \vee s_i, \quad \neg s_{i-1} \vee s_i, \quad \neg x_i \vee \neg s_{i-1}.$$

Mệnh đề cuối cấm x_i khi tiền tố trước đó đã có một giá trị đúng. Kích thước tuyến tính theo n [72].

3.3.3. Biểu diễn tích và biểu diễn chỉ huy

Product encoding (biểu diễn tích) đặt biến vào ma trận rồi dùng AMO theo hàng và cột. *Commander encoding* (biểu diễn chỉ huy) chia biến thành nhóm, tạo một biến đại diện cho mỗi nhóm rồi biểu diễn AMO trên các đại diện. Hai cách này tạo tính cục bộ theo khối và phù hợp với miền vừa hoặc lớn [21].

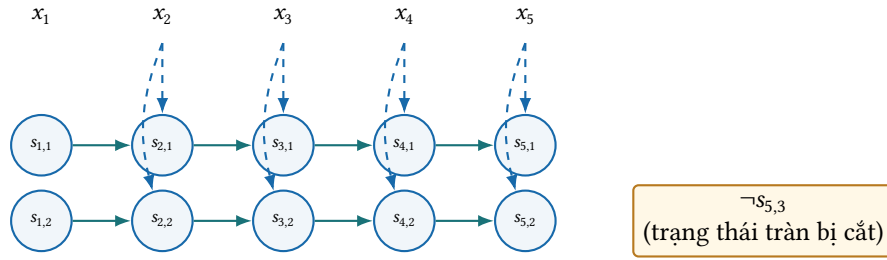
Bảng 3.1.: So sánh các bộ biểu diễn AMO cơ bản.

Phép biểu diễn AMO	Biến phụ	Mệnh đề	Điểm mạnh
Từng cặp	0	$O(n^2)$	Nhị phân, đơn giản
Tuần tự	$O(n)$	$O(n)$	Lan truyền theo tiền tố
Tích	$O(\sqrt{n})$	$O(n)$	Cân bằng hai chiều
Chỉ huy	$O(n/g)$	Phụ thuộc nhóm g	Cục bộ, dễ kết hợp

3.4. Sequential Counter cho AMK

Đặt

$$s_{i,j} = 1 \iff \sum_{q=1}^i x_q \geq j, \quad 1 \leq j \leq k.$$



Hình 3.2.: Luồng trạng thái của Sequential Counter cho $n = 5, k = 2$. Mũi tên liền truyền ngưỡng; mũi tên đứt đưa đầu vào mới vào hệ thức truy hồi.

Bộ đếm cập nhật theo hệ thức truy hồi

$$s_{i,j} \leftrightarrow s_{i-1,j} \vee (x_i \wedge s_{i-1,j-1}),$$

với $s_{i,0} = \top$. Cận AMK cấm trạng thái tràn:

$$\neg x_i \vee \neg s_{i-1,k}.$$

Số thanh ghi và mệnh đề là $O(nk)$. Với k nhỏ, bộ đếm tạo các quan hệ kéo theo cục bộ và cho phép đọc nhiều ngưỡng tiền tố [72].

Mệnh đề 3.1 (Ngữ nghĩa của bộ đếm). Nếu các hệ thức truy hồi được biểu diễn hai chiều thì, với mọi i, j ,

$$s_{i,j} = 1 \iff \sum_{q=1}^i x_q \geq j.$$

Chứng minh. Quy nạp theo i . Tiền tố i đạt j khi tiền tố $i-1$ đã đạt j , hoặc $x_i = 1$ và tiền tố trước đạt $j-1$. Đây chính là hệ thức truy hồi. $\square$

Xét $x_1 + \dots + x_5 \leq 2$. Sau khi $x_1 = x_3 = 1$, bộ đếm suy ra

$$s_{3,2} = 1.$$

Mệnh đề chống tràn tại $i = 4$,

$$\neg x_4 \vee \neg s_{3,2},$$

trở thành mệnh đề đơn vị và suy ra $\neg x_4$. Tương tự, tính đơn điệu của tiền tố đưa $s_{3,2}$ đến $s_{4,2}$, rồi mệnh đề chống tràn tại $i = 5$ suy ra $\neg x_5$. Bộ đếm không cần đợi đầu vào đúng thứ ba mới báo xung đột; cấu trúc sẽ loại mọi vị trí còn lại ngay khi cận đã đầy.

3.5. Totalizer

Totalizer xây dựng một cây nhị phân. Mỗi nút trả về một vectơ đơn phân

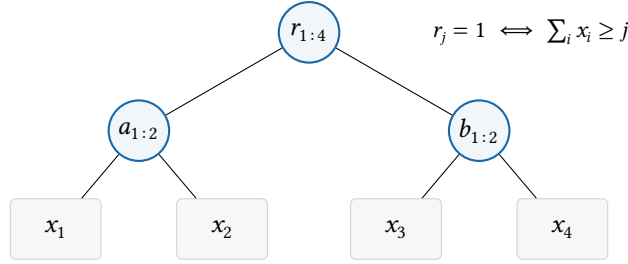
$$o_j = 1 \iff \text{tổng trong cây con} \geq j.$$

Khi ghép hai nút con có đầu ra a, b , nút cha r thỏa

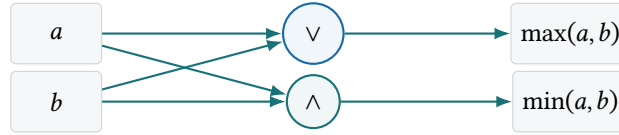
$$a_p \wedge b_q \rightarrow r_{p+q}.$$

Chiều ngược bảo đảm đặc tả đầy đủ nếu các đầu ra được dùng bên ngoài. Cận $\sum x_i \leq k$ chỉ cần $\neg o_{k+1}$.

Totalizer là lựa chọn tự nhiên khi:



Hình 3.3.: Totalizer cân bằng: mỗi nút trả về các ngưỡng của tổng trong cây con.



Hình 3.4.: Bộ so sánh Boolean là thành phần cơ sở của Sorting Network và mạng đếm.

- cần nhiều ngưỡng trên cùng một tập biến;
- cần được thay đổi tăng dần;
- cần ràng buộc đúng K bằng $o_k \wedge \neg o_{k+1}$.

Totalizer cân bằng tạo tính cục bộ theo cây; kích thước có thể được cắt ở $k + 1$ thay vì giữ toàn bộ tổng [9].

3.6. Mạng đếm

Sorting Network sắp các đầu vào Boolean thành các đầu ra đơn phân:

$$o_1 \geq o_2 \geq \dots \geq o_n, \quad o_j = 1 \iff \sum_i x_i \geq j.$$

Bộ so sánh

$$(a, b) \mapsto (a \vee b, a \wedge b)$$

được chuyển đổi theo phương pháp Tseitin. Cận AMK là $\neg o_{k+1}$. Mạng có cấu trúc đều, độ sâu đa lôgarit và thuận lợi cho lan truyền song song [5, 10].

So với Sequential Counter, mạng dùng nhiều mệnh đề hơn khi k nhỏ nhưng có độ sâu suy diễn thấp hơn và các đầu ra toàn cục rõ ràng.

3.7. Biểu diễn bằng BDD

Binary decision diagram (biểu đồ quyết định nhị phân, BDD) biểu diễn trạng thái tổng tích lũy. Nút ở mức i và trạng thái s phân nhánh theo x_i . Các trạng thái vượt k đi vào nút kết thúc sai. Một biến phụ cho mỗi nút BDD cùng các mệnh đề liên kết theo hai cung tạo thành phép biểu diễn [2, 20].

Ưu điểm của BDD:

- xử lý tự nhiên các ràng buộc giả Boolean có trọng số;
- có thể đạt sức lan truyền mạnh;
- hợp nhất các trạng thái tương đương.

Nhược điểm là kích thước phụ thuộc thứ tự biến và số trạng thái tổng khác nhau.

3.8. Ràng buộc đúng K và đầu ra chia sẻ

EXK có thể viết thành AMK và ALK, nhưng dùng hai bộ đếm độc lập sẽ lập trạng thái. Nếu bộ biểu diễn trả về các đầu ra đơn phân o_j , chỉ cần

$$o_k \wedge \neg o_{k+1}.$$

Sequential Counter cũng có thể chia sẻ thành ghi: biên $s_{n,k}$ bảo đảm ít nhất k , còn các mệnh đề chống tràn bảo đảm nhiều nhất k .

Tách CounterDefinition khỏi Bound. Cùng một cấu trúc đầu ra có thể phục vụ AMK, ALK, EXK và vòng giải tăng dần. Đây là bước đầu của tư duy mô-đun được phát triển ở Phần II.

3.9. Pseudo-Boolean có trọng số

Ràng buộc tổng quát

$$\sum_{i=1}^n a_i x_i \leq B, \quad a_i \in \mathbb{Z}_{\geq 0},$$

không còn là cardinality khi trọng số khác nhau. Các hướng chính gồm:

- biểu diễn bằng bộ cộng hoặc mạch cho biểu diễn nhị phân của tổng;
- BDD cho các trạng thái tổng có thể đạt;
- Totalizer tổng quát gom và cắt các tổng;
- phân rã bằng Sorting Network hoặc mạng đếm sau khi tách trọng số;
- thư viện kết hợp như PBLib chọn bộ biểu diễn theo cấu trúc [34, 68].

Nếu hàm mục tiêu thay cận nhiều lần, các đầu ra hoặc giao diện kích hoạt quan trọng hơn vài phần trăm kích thước ở một cận duy nhất.

3.9.1. Totalizer tổng quát

Generalized totalizer (Totalizer tổng quát) thay vectơ đơn phân liên tiếp bằng tập các tổng có thể đạt tại mỗi nút. Khi ghép hai nút có tập tổng A, B , nút cha giữ các giá trị

$$\min(a + b, B + 1), \quad a \in A, b \in B,$$

trong đó $B + 1$ gom mọi trạng thái tràn. Phép biểu diễn đặc biệt hiệu quả khi trọng số có cấu trúc và số tổng phân biệt sau phép cắt nhỏ hơn tổng số học thô [46]. Hình dạng cây không còn là chi tiết vô hại: hai cách ghép các lá có thể tạo số trạng thái trung gian rất khác nhau.

3.9.2. Khai thác các nhóm AMO trong PB

Trong nhiều mô hình, các hạng tử của ràng buộc PB đến từ một biến miền hữu hạn: trong mỗi nhóm, nhiều nhất một literal có thể đúng. Nếu trải phẳng mọi hạng tử như các biến độc lập, bộ biểu diễn sẽ bỏ mất thông tin này. Các phép biểu diễn PB(AMO) đưa nhóm AMO trực tiếp vào *multi-valued decision diagram* (biểu đồ quyết định đa trị, MDD), Totalizer tổng quát, Sequential Counter có trọng số hoặc Totalizer theo môđun [16]. Hai lợi ích có thể xuất hiện:

- ít trạng thái tổng hơn, vì tổ hợp chọn hai hạng tử cùng nhóm không bao giờ đạt được;

- lan truyền dùng được cả cận PB lẫn tính loại trừ lẫn nhau trong nhóm.

Đây là một ví dụ tổng quát của *structure-aware encoding* (phép biểu diễn có xét cấu trúc): trước khi chọn bộ biểu diễn cho một biểu thức tuyến tính, cần giữ nguồn gốc cấu trúc của các hạng tử thay vì chỉ truyền vào mảng hệ số.

3.10. Tù lựa chọn thủ công đến lựa chọn theo bài toán thử

Quy tắc “ k nhỏ thì dùng Sequential Counter, nhiều cận thì dùng Totalizer” là một đối chứng hữu ích, nhưng chưa bao quát tương tác giữa hệ số, nhóm AMO, độ rộng miền và bộ giải. Nghiên cứu về *algorithm selection* (lựa chọn thuật toán) đã dùng các đặc trưng của ràng buộc và bài toán thử để chọn giữa nhiều phép biểu diễn SAT, kể cả trên những lớp bài toán chưa xuất hiện trong dữ liệu huấn luyện [87]. Kết quả này thay đổi cách xây dựng thư viện bộ biểu diễn: mỗi mô-đun cần công bố đủ siêu dữ liệu để một bộ chọn ra quyết định.

Bảng 3.2.: Nhóm đặc trưng để lựa chọn bộ biểu diễn theo bài toán thử.

Nhóm đặc trưng	Ví dụ	Tác động tiềm năng
Kích thước	n, k, B , số hạng tử	Loại bộ biểu diễn chắc chắn quá lớn
Trọng số	Giá trị lớn nhất, ước chung lớn nhất, số tổng phân biệt	Không gian trạng thái BDD/GT
Cấu trúc	Nhóm AMO, hệ số lặp	Khả năng nén trạng thái
Giao diện	Số cận, thêm đầu vào, hiện thực hóa (reification)	Giải tăng dần và kết hợp mô-đun
Mô hình	Mật độ, đối xứng, đồ thị ràng buộc	Tương tác với học mệnh đề

Bộ chọn không loại bỏ nhu cầu chứng minh tính đúng: công cụ chỉ lựa chọn giữa các mô-đun đã được kiểm chứng. Việc đánh giá cũng phải tách các họ dùng để huấn luyện khỏi các họ dùng để kiểm thử; nếu không, bộ chọn có thể chỉ nhận dạng tên bộ kiểm chuẩn thay vì học cấu trúc của phép biểu diễn.

3.11. Bản đồ lựa chọn

Bảng 3.3.: Gợi ý lựa chọn bộ biểu diễn ràng buộc đếm/PB.

Tình huống	Lựa chọn khởi đầu	Lý do
AMO, n nhỏ	Từng cặp	Không biến phụ, mệnh đề nhị phân
AMO/AMK, k nhỏ	Tuần tự	$O(nk)$, lan truyền theo tiền tố
Nhiều ngưỡng	Totalizer/mạng đếm	Tái sử dụng đầu ra
Đúng K	Một vectơ đầu ra chung	Không lặp AMK/ALK
PB có trọng số, ít trạng thái	BDD	Hợp nhất trạng thái tổng
PB có trọng số tổng quát	PBLib/kết hợp	Chọn theo bài toán thử
PB có nhóm AMO	Bộ biểu diễn PB(AMO)	Không hiện thực hóa (reification) trạng thái bất khả
Cận thay đổi tăng dần	Totalizer/đầu ra thứ tự	Giao diện cận ổn định

Đây là cấu hình khởi đầu; Chương 4 trình bày cách đánh giá bộ biểu diễn trên phân phối bài toán

mục tiêu; Chương 5–6 phát triển các bộ đếm khi nhiều ràng buộc chia sẻ cấu trúc.

3.12. Bài học thiết kế

1. Chọn bộ biểu diễn theo cận và đầu ra cần đọc, không chỉ theo tên ràng buộc.
2. Biểu diễn AMO từng cặp vẫn là đối chứng mạnh khi miền nhỏ.
3. Sequential Counter phù hợp với ngưỡng nhỏ và cấu trúc tiền tố.
4. Totalizer/mạng đếm thuận lợi khi cần nhiều cận hoặc giải tăng dần.
5. Ràng buộc đúng K nên chia sẻ một cấu trúc tổng nếu có thể.

Câu hỏi nghiên cứu

1. Dựng Sequential Counter cho $n = 5, k = 2$ và liệt kê các thanh ghi.
2. So sánh EXK bằng hai bộ đếm độc lập với một Totalizer chung.
3. Với trọng số $(1, 2, 3, 5)$, vẽ các trạng thái BDD cho cận $B = 5$.

4 Thiết kế thực nghiệm cho phép biểu diễn SAT

TÓM TẮT CHƯƠNG

Đánh giá trên bộ kiểm chuẩn không chỉ trả lời bộ biểu diễn nào nhanh hơn. Một thực nghiệm tốt còn phải giải thích hiệu ứng đến từ kích thước, sức lan truyền, bộ giải, đặc trưng hay cấu trúc của bài toán thử.

4.1. Thực nghiệm là một phép đo có mô hình

Một bảng thời gian chạy không tự nói bộ biểu diễn nào tốt hơn. Quan sát được tạo bởi cả chuỗi

bài toán thử → mô hình → bộ biểu diễn → bộ tiền xử lý → cấu hình bộ giải → phần cứng.

Thay bất kỳ mắt xích nào cũng có thể đổi thứ hạng. Vì vậy kết luận cần nêu rõ *population* (quần thể mà kết luận hướng đến): một họ bài toán thử, một miền tham số, một bộ giải hay một danh mục bộ giải. Cần phân biệt *internal validity* (hiệu lực nội tại: hiệu ứng thật sự do yếu tố đang xét) với *external validity* (hiệu lực khái quát: hiệu ứng có còn giữ ngoài tập kiểm chuẩn hay không).

Trong nghiên cứu phép biểu diễn, ba loại bằng chứng bổ sung nhau:

- **bằng chứng cơ chế:** công thức đếm, vết lan truyền và xung đột nhỏ giải thích vì sao hiệu ứng xuất hiện;
- **bằng chứng có kiểm soát:** *ablation* (thí nghiệm phân tích thành phần (*ablation study*)) giữ cố định các yếu tố khác;
- **bằng chứng so sánh:** đánh giá trên nhiều họ bài toán và nhiều bộ giải để kiểm tra độ bền của kết luận.

Một kết quả có giá trị tham khảo cao cần ít nhất hai loại; thời gian chạy đơn lẻ chỉ là bằng chứng so sánh yếu nếu thiếu cơ chế và biến kiểm soát.

4.2. Bắt đầu từ một câu hỏi nhân quả

Một nghiên cứu về phép biểu diễn thường có một trong bốn câu hỏi:

1. phép biểu diễn môi giảm kích thước bao nhiêu;
2. sức lan truyền thay đổi như thế nào;
3. thời gian chạy và số bài giải được thay đổi ra sao;



Hình 4.1.: Lộ trình lập luận của chương: từ Câu hỏi, qua Ma trận thiết kế, Bộ kiểm chuẩn và Thuộc đo, đến Phân tích.

4. một mô-đun có tương tác với phá vỡ đối xứng, giải tăng dần hoặc bộ giải không.

Câu hỏi càng rõ, ma trận thiết kế càng nhỏ. Nếu mục tiêu là đo bộ đếm dùng chung, hãy giữ cố định bộ giải, tiền xử lý, chiến lược tìm kiếm và quy trình xử lý hàm mục tiêu; chỉ thay mô-đun bộ đếm. Nếu đồng thời thay bộ giải và cách phá vỡ đối xứng, kết quả không phân định được đóng góp riêng lẻ của từng thành phần.

Một đặc trưng không có hiệu ứng độc lập khi làm thay đổi giao diện của phép biểu diễn. Trong trường hợp đó, dùng cấu hình phép biểu diễn $\times$ đặc trưng làm đơn vị so sánh.

4.3. Ma trận thiết kế

Một ma trận điển hình gồm:

bộ biểu diễn $\times$ bộ giải $\times$ đặc trưng $\times$ bài toán thử $\times$ random seed.

Ở đây *random seed* (hạt giống ngẫu nhiên) là giá trị khởi tạo bộ sinh số giả ngẫu nhiên; giữ lại giá trị này giúp tái lập đúng chuỗi lựa chọn ngẫu nhiên của một lần chạy. Không nhất thiết chạy tích Descartes đầy đủ. Có thể dùng hai giai đoạn:

1. **sàng lọc**: chọn vài bộ giải đại diện và miền tham số rộng;
2. **xác nhận**: giữ các cấu hình có triển vọng, tăng số bài toán thử và số hạt giống ngẫu nhiên.

Mọi cấu hình phải dùng cùng ngữ nghĩa và bộ kiểm tra. Nếu một bộ biểu diễn có tiền xử lý riêng, thời gian tiền xử lý phải được tính vào tổng thời gian chạy hoặc được báo thành một cột riêng.

4.4. Bộ kiểm chuẩn và phân tầng

Một bộ kiểm chuẩn nên có siêu dữ liệu cấu trúc: số biến miền, mật độ ràng buộc, độ rộng của số, ngưỡng, lớp đối xứng, khoảng $LB-UB$, hoặc các đặc trưng riêng của bài toán. Phân tầng theo những đặc trưng này giúp giải thích tại sao một phép biểu diễn thắng trong một vùng.

Ba nguồn bài toán thử thường dùng:

- bài toán chuẩn từ văn liệu;
- bài toán thực tế có nguồn gốc rõ ràng;
- bài toán sinh tổng hợp để quét có kiểm soát một tham số.

Các bài toán sinh tổng hợp cho phép quan sát điểm chuyển theo n, k, w , mật độ hoặc độ rộng miền; dữ liệu thực cung cấp bối cảnh sử dụng, còn bài toán chuẩn tạo khả năng so sánh với kết quả trước. Báo riêng ba nguồn giúp người đọc hiểu đúng vai trò của từng tập.

4.5. Tách tinh chỉnh và đánh giá

Tùy chọn bộ giải, độ rộng khối, kích thước nhóm và ngưỡng biểu diễn là các siêu tham số. Nếu chọn chúng trên chính tập kiểm thử, thời gian chạy báo cáo sẽ lạc quan quá mức. Một quy trình đơn giản:

1. chia bộ kiểm chuẩn thành tập tinh chỉnh và tập đánh giá;
2. khóa cấu hình sau khi tinh chỉnh;

3. chỉ dùng tập đánh giá cho kết luận cuối;
4. nếu dữ liệu ít, dùng *cross-validation* (kiểm định chéo) theo họ thay vì chia ngẫu nhiên từng bài toán. Tình hình theo họ tránh việc các biến thể gần như trùng nhau xuất hiện ở cả hai tập.

4.6. Các thước đo

4.6.1. Kích thước và xây dựng

Ghi ít nhất:

$$n_v, \quad n_c, \quad n_t, \quad \#binary, \quad \#ternary, \quad t_{\text{encode}}, \quad \text{bộ nhớ cực đại.}$$

Thời gian của bộ giải nên đo từ sau khi dựng công thức; đồng thời báo thời gian toàn trình nếu bộ sinh công thức chiếm một phần đáng kể.

4.6.2. Quá trình tìm kiếm

Số quyết định, lượt lan truyền, xung đột, lần khởi động lại và mệnh đề đã học giúp giải thích cơ chế tạo ra thời gian chạy quan sát được:

$$\text{lượt lan truyền/xung đột,} \quad \text{xung đột/giây,} \quad \text{bộ nhớ/biến.}$$

4.6.3. Right-censored runtime: thời gian chạy có kiểm duyệt

Khi một lần chạy hết thời hạn, ta chỉ biết thời gian cần thiết lớn hơn giới hạn; dữ liệu như vậy được gọi là *right-censored* (dữ liệu bị giới hạn thời gian). Vì vậy, thống kê thời gian chạy phải tính cả bài đã giải và bài hết thời hạn bằng một trong các cách:

- số bài giải được tại một giới hạn thời gian cố định;
- PAR-2 hoặc PAR-10, tức thời gian trung bình có phạt khi hết thời hạn;
- *cactus plot* (biểu đồ Cactus (Cactus plot));
- *performance profile* (hồ sơ hiệu năng);
- thời gian đạt mục tiêu cho bài toán tối ưu.

Hồ sơ hiệu năng của Dolan–Moré chuẩn hóa thời gian chạy theo cấu hình tốt nhất trên từng bài toán thử. Với tỷ số

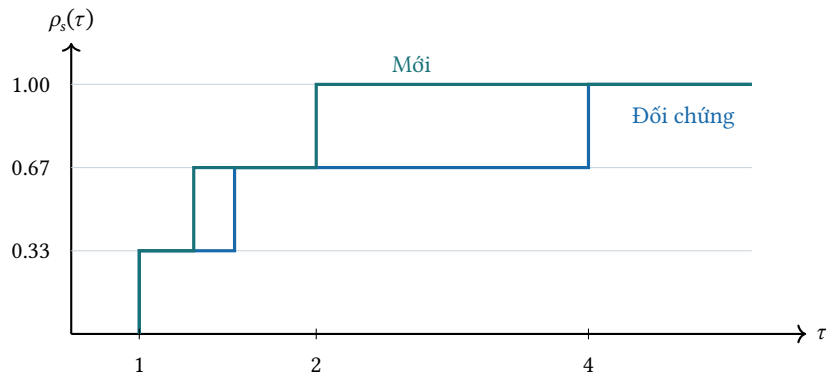
$$r_{p,s} = \frac{t_{p,s}}{\min_{s'} t_{p,s'}},$$

đường

$$\rho_s(\tau) = \frac{1}{|P|} |\{p \in P \mid r_{p,s} \leq \tau\}|$$

cho biết tỷ lệ bài mà cấu hình s nằm trong hệ số τ so với tốt nhất [31]. Giá trị tại $\tau = 1$ đo số lần thắng; phần đuôi đo độ bền. Bài hết thời hạn phải được gán một tỷ số lớn nhất nhất quán và được mô tả rõ.

Một cấu hình giải thêm các bài khó có thể có tổng thời gian chạy lớn hơn cấu hình hết thời hạn sớm. Vì vậy không xếp hạng bằng một cột thời gian đơn lẻ.



Hình 4.2.: Hồ sơ hiệu năng (performance profile): phần đầu cho thấy số bài giải nhanh nhất, phần đuôi cho độ bền so với cấu hình tốt nhất trên từng bài toán thử.

Với giới hạn 30 giây, ta có:

	p_1	p_2	p_3
Đối chứng	2	5	hết hạn
Mới	1	10	20

Đối chứng nhanh hơn trên p_2 , nhưng chỉ giải 2/3 bài; cấu hình mới giải cả ba. Nếu PAR-2 gán 60 giây cho lần chạy hết thời hạn:

$$\text{PAR2(Đối chứng)} = \frac{2 + 5 + 60}{3} = 22.33, \quad \text{PAR2(Mới)} = \frac{1 + 10 + 20}{3} = 10.33.$$

Trung bình chỉ trên bài đã giải sẽ cho đối chứng 3.5 giây và cấu hình mới 10.33 giây, từ đó tạo kết luận ngược vì đã bỏ chính bài khó nhất.

4.6.4. Biểu đồ Cactus (Cactus plot) và biểu đồ phân tán

Với một cấu hình s , sắp tăng thời gian của các bài đã giải:

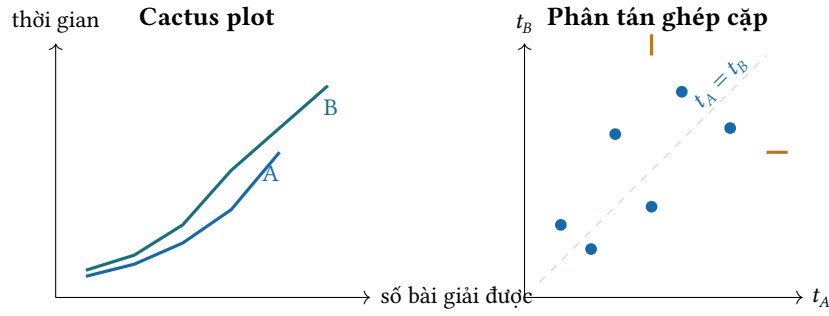
$$t_{(1),s} \leq t_{(2),s} \leq \dots \leq t_{(q_s),s}.$$

Biểu đồ Cactus (Cactus plot) đặt số bài đã giải j trên trục ngang và $t_{(j),s}$ trên trục dọc. Đường đi xa hơn sang phải biểu thị độ bền; đường thấp hơn biểu thị giải nhanh hơn ở cùng số bài đã giải. Đường dừng ở q_s , nên các bài hết thời hạn không bị âm thầm loại khỏi kết luận.

Biểu đồ phân tán dùng so sánh ghép cặp: mỗi điểm ứng với cùng một bài toán thử, có tọa độ $(t_{p,A}, t_{p,B})$, thường trên thang lôgarit. Điểm phía trên đường chéo $t_A = t_B$ nghĩa là A nhanh hơn; điểm phía dưới nghĩa là B nhanh hơn. Bài hết thời hạn cần được đặt ở biên T bằng ký hiệu riêng hoặc mũi tên, và SAT/UNSAT nên dùng dấu điểm khác nhau. Biểu đồ phân tán cho biết *bài toán nào* tạo khác biệt; biểu đồ Cactus (Cactus plot) cho biết *toàn tập* được bao phủ đến đâu.

4.7. SAT và UNSAT là hai chế độ khác nhau

Các bài toán SAT cần tìm một nghiệm làm chứng; các bài toán UNSAT cần đóng toàn bộ không gian tìm kiếm. Một phép biểu diễn có thể mạnh ở SAT nhưng yếu ở UNSAT hoặc ngược lại. Bảng kết quả nên



Hình 4.3.: Biểu đồ Cactus (Cactus plot) (trái) giữ số bài giải được trong hình; biểu đồ phân tán (phải) giữ ghép cặp theo bài toán thử. Các dấu ở biên minh họa lần chạy hết thời hạn của một cấu hình.

tách:

$$\#SAT, \#UNSAT, t_{SAT}, t_{UNSAT}.$$

Trong tối ưu, các lần gọi bộ giải gần giá trị tối ưu thường có cả hai trạng thái: mô hình tại cận khả thi và UNSAT tại cận tốt hơn. Vì thế, hồ sơ theo toàn bộ dãy cận chứa nhiều thông tin hơn thời gian chạy cuối cùng.

4.8. Random seed và biến thiên giữa các lần chạy

Random seed, được gọi là hạt giống ngẫu nhiên sau lần giới thiệu đầu, không phải dữ liệu của bài toán; tham số này là giá trị khởi tạo (seed) cho bộ sinh số giả ngẫu nhiên của thuật toán. CDCL có yếu tố ngẫu nhiên trong phá hòa và tìm kiếm. Với bài toán khó, cần chạy nhiều hạt giống ngẫu nhiên và báo trung vị, các phân vị hoặc khoảng tin cậy. So sánh ghép cặp phải dùng cùng bài toán, cùng chỉ số hạt giống ngẫu nhiên và cùng giới hạn thời gian, bộ nhớ.

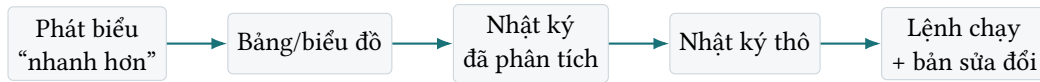
Nếu bộ biểu diễn là tất định nhưng bộ giải có yếu tố ngẫu nhiên, hạt giống ngẫu nhiên thuộc về bộ giải. Nếu bộ sinh dùng thứ tự ngẫu nhiên, cần tách hạt giống ngẫu nhiên của bộ sinh khỏi hạt giống ngẫu nhiên của bộ giải.

4.9. So sánh ghép cặp và độ lớn hiệu ứng

Mỗi bài toán thử tạo một cặp quan sát tự nhiên giữa hai bộ biểu diễn. Vì phân bố thời gian chạy thường lệch và có dữ liệu hết thời hạn, chỉ báo trung bình và một kiểm định trên các lần chạy độc lập là chưa phù hợp. Nên báo đồng thời:

- số lần thắng/hòa/thua trên cùng bài toán và cùng hạt giống ngẫu nhiên;
- trung vị của lôgarit hệ số tăng tốc $\log(t_{\text{base}}/t_{\text{new}})$;
- khoảng bất định bằng *bootstrap* (lấy mẫu lại) theo *họ*, không chỉ theo từng bài toán;
- chênh lệch số bài giải được tại nhiều mốc thời gian.

Độ lớn hiệu ứng quan trọng hơn một giá trị p đứng riêng. Một bộ biểu diễn nhanh hơn 3% trên hàng nghìn bài dễ nhưng làm mất năm bài khó có ý nghĩa khác với bộ biểu diễn chỉ giảm ít trung vị nhưng mở rộng vùng giải được. Với tối ưu, nên so cả thời gian đạt nghiệm tốt nhất đã biết, thời gian hoàn tất chứng minh và khoảng cách tối ưu theo thời gian.



Mỗi mũi tên phải có chương trình, mã kiểm tra hoặc siêu dữ liệu để truy vết.

Hình 4.4.: Chuỗi truy vết tối thiểu của một phát biểu thực nghiệm.

4.10. Phân tích theo vùng cấu trúc

Một kết luận toàn cục dễ che mất điểm chuyển ưu thế. Sequential Counter có thể tốt khi k/n nhỏ; mạng đếm có thể cải thiện khi cần nhiều ngưỡng; Order Encoding có thể thắng khi các khoảng cấm rộng. Vì vậy, trước khi vẽ kết quả, cần xác định trước các tầng theo đặc trưng có ý nghĩa thuật toán:

$\frac{k}{n}$, độ chồng lấn của số, độ rộng miền, mật độ ràng buộc, quy mô đối xứng.

Sau đó dùng bản đồ nhiệt hoặc bảng theo tầng để tìm *miền ưu thế* thay vì chỉ tìm “người thắng”. Cách phân tích này cũng cung cấp dữ liệu cho bộ chọn phép biểu diễn ở Chương 3.

4.11. Kiểm đúng trước khi đo nhanh

Trước khi chạy bộ kiểm chuẩn lớn:

1. kiểm thử vết cạn trên các bài toán nhỏ;
2. kiểm thử vi sai giữa hai bộ biểu diễn;
3. giải mã và kiểm tra mọi mô hình SAT dùng trong bảng;
4. tính lại hàm mục tiêu từ nghiệm, không đọc trực tiếp chi phí nội bộ;
5. giữ một số công thức CNF và chứng minh nhỏ làm phép thử hồi quy.

Với n nhỏ, bộ liệt kê sinh mọi phép gán cho biến chính. So sánh phép chiếu của các mô hình CNF với tập nghiệm miền. Cách này phát hiện lỗi biên hiệu quả hơn việc chỉ chạy các bài toán kiểm chuẩn lớn.

4.12. Gói tái lập và khả năng truy vết

Artifact (gói phần mềm–dữ liệu tái lập) tối thiểu gồm:

- mã nguồn và mã định danh bản sửa đổi chính xác;
- bản kê bài toán thử và mã kiểm tra toàn vẹn;
- môi trường cùng phiên bản bộ giải;
- lệnh chạy và giới hạn tài nguyên;
- nhật ký thô, bộ phân tích nhật ký và chương trình tạo bảng;
- bộ giải mã và bộ kiểm tra.

Gói tái lập tạo một đường đi kiểm tra được từ phát biểu đến bằng chứng thô.

Số liệu thực nghiệm có thể được kiểm tra ở ba tầng:

1. **báo cáo:** bài viết mô tả tập bài toán thử, giới hạn thời gian và các cấu hình đối chứng;
2. **tái tạo:** mã nguồn, nhật ký và dữ liệu cho phép dựng lại các bảng và biểu đồ;

3. **tái lập độc lập:** một môi trường khác thực hiện lại thí nghiệm theo cùng thiết kế.

Tăng báo cáo hỗ trợ việc diễn giải kết quả; tăng tái tạo kiểm tra đường đi từ nhật ký đến bảng; tăng tái lập độc lập đánh giá độ bền khi môi trường thực thi thay đổi.

4.13. Mẫu quy trình thực nghiệm

Câu hỏi. Mô-đun nào được đánh giá?

Giả thuyết. Kỳ vọng tác động lên kích thước, sức lan truyền hay thời gian chạy?

Yếu tố. Bộ biểu diễn, bộ giải, đặc trưng, họ bài toán, hạt giống ngẫu nhiên.

Biến kiểm soát. Ngữ nghĩa, bộ kiểm tra, cận, tiền xử lý, phần cứng.

Thuốc đo. Kích thước, thời gian dựng, bộ đếm tìm kiếm, số bài giải được/PAR/hồ sơ.

Kiểm đúng. Kiểm thử vết cận, kiểm thử vi sai và bộ kiểm tra mô hình.

Đầu ra. Nhật ký thô, biểu đồ, bảng tổng hợp và bản kê gói tái lập.

4.14. Bài học thiết kế

1. Đánh giá thực nghiệm bắt đầu từ một câu hỏi, không bắt đầu từ một bảng cấu hình.
2. Tách tinh chỉnh khỏi đánh giá và phân tầng theo họ bài toán.
3. Báo cả kích thước, bộ đếm tìm kiếm và thời gian chạy có dữ liệu hết hạn.
4. Tách SAT/UNSAT và dùng nhiều hạt giống ngẫu nhiên ở vùng khó.
5. Bộ kiểm tra mô hình là điều kiện trước của mọi bảng hiệu năng.

Câu hỏi nghiên cứu

1. Thiết kế thí nghiệm phân tích thành phần (ablation study) 2×2 cho bộ biểu diễn và phá vỡ đối xứng.
2. Với ba lần hết thời hạn trong mười bài toán thử, tính PAR-2 và so với trung bình trên các bài đã giải.
3. Lập một bản kê gói tái lập tối thiểu cho một thí nghiệm SAT tăng dần.

PHẦN II

**Tái sử dụng cấu trúc trong phép
biểu diễn SAT**

5 Shared Counter cho các cửa sổ chồng lấn

TÓM TẮT CHƯƠNG

Khi một ràng buộc trượt trên chuỗi, hai cửa sổ kề nhau chia sẻ gần toàn bộ biến. Biểu diễn từng cửa sổ độc lập sẽ lặp trạng thái; Shared Counter (bộ đếm chia sẻ) biến phần giao đó thành một mô-đun có thể tái sử dụng.

5.1. SEQUENCE trong văn liệu về ràng buộc

Ràng buộc cửa sổ thuộc một họ lâu đời hơn các Shared Counter. SEQUENCE yêu cầu mọi đoạn liên tiếp độ dài cố định có số biến nhận một giá trị trong khoảng $[l, u]$. Dạng ràng buộc này xuất hiện trong bài toán lập lịch ca, xếp thứ tự xe, và xác định cỡ lô và các mô hình chuỗi. Vấn đề cốt lõi không chỉ là số literal đúng trong một cửa sổ, mà còn là sự phụ thuộc giữa toàn bộ các cửa sổ chồng lấn.

Văn liệu đã phát triển ít nhất bốn cách nhìn:

Phân rã theo cửa sổ

Tạo một ràng buộc đếm cho mỗi cửa sổ. Cách này có tính mô-đun và dễ dùng, nhưng nếu các trạng thái phụ không được chia sẻ thì phần giao bị biểu diễn nhiều lần.

Tổng tiền tố

Đặt $S_i = \sum_{j \leq i} x_j$, khi đó mỗi cửa sổ là $l \leq S_{i+w} - S_i \leq u$. Mọi cửa sổ chia sẻ một dãy trạng thái tiền tố.

Ôtômát/ràng buộc chính quy

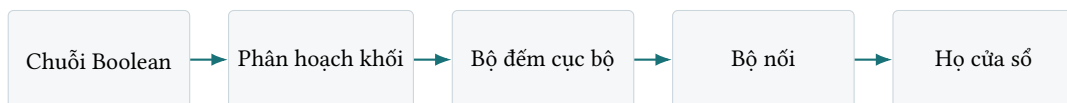
Trạng thái ghi đủ lịch sử cần thiết để quyết định ký hiệu tiếp theo. Một ràng buộc REGULAR có thể mô hình hóa bài toán kiểm tra chuỗi thuộc một ngôn ngữ hữu hạn trạng thái hay không [65].

Luồng hoặc đồ thị

Biểu diễn các hiệu tiền tố và cận cửa sổ bằng mạng luồng, từ đó xây dựng bộ lan truyền toàn cục.

So sánh có hệ thống các phép biểu diễn SEQUENCE cho thấy cách phân rã, ôtômát và phương pháp dựa trên tổng khác nhau rõ về kích thước và mức nhất quán [19]. Các bộ lan truyền dựa trên luồng tiếp tục khai thác cấu trúc toàn cục để đạt khả năng lọc mạnh [56]; những khung tổng quát hơn còn xem ràng buộc trên chuỗi như tổ hợp của một *chữ ký* và một *đặc trưng* cần đếm [61].

Bảng 5.1.: Các cách tái sử dụng trạng thái trong ràng buộc cửa sổ.



Hình 5.1.: Lộ trình lập luận của chương: từ Chuỗi Boolean, qua Phân hoạch khối, Bộ đếm cục bộ và Bộ nối, đến Họ cửa sổ.

Họ	Trạng thái chia sẻ	Điểm mạnh	Chi phí/rủi ro
Mỗi cửa sổ	Không hoặc ít	Đơn giản, dùng bộ biểu diễn sẵn	Lập trạng thái phụ
Tổng tiền tố	Tổng từ đầu chuỗi	Mọi cửa sổ là hiệu hai tiền tố	Miền tổng toàn cục
Ôtômát	Hậu tố/lich sử	Xử lý mẫu phức tạp	Bùng nổ trạng thái theo w
Luồng	Nút và cung	Lọc toàn cục	Bộ lan truyền/phép biểu diễn phức tạp
Bộ đếm theo khối	Tiền tố/hậu tố trong khối	Cục bộ và tái sử dụng có kiểm soát	Cần chọn phân hoạch

Shared Counter theo khối của chương này nằm giữa hai cực. Phương pháp không tạo một trạng thái toàn cục cho mọi tiền tố, nhưng cũng không xây bộ đếm độc lập cho từng cửa sổ. Phân hoạch quyết định phạm vi tái sử dụng, còn bộ nối giữ ngữ nghĩa của từng cửa sổ.

5.2. Họ ràng buộc của sổ

Cho chuỗi

$$\Omega = \langle x_1, \dots, x_n \rangle.$$

Cửa sổ độ rộng w bắt đầu tại i :

$$W_i(w) = \langle x_i, \dots, x_{i+w-1} \rangle.$$

AMK bậc thang là họ

$$\sum_{x \in W_i(w)} x \leq k \quad i = 1, \dots, n - w + 1.$$

Các ràng buộc chuỗi tổng quát cho phép cả cận dưới và cận trên:

$$\ell \leq \sum_{x \in W_i(w)} x \leq u.$$

Nếu mỗi cửa sổ dùng một Sequential Counter riêng, kích thước xấp xỉ

$$O((n - w + 1)wk).$$

Trong khi đó, hai cửa sổ kề nhau chỉ khác một biến đi ra và một biến đi vào. Mục tiêu của phép biểu diễn chia sẻ là đưa chi phí về gần

$$O(nk)$$

hoặc $O(nk + n_c)$, với n_c là số bộ nối.

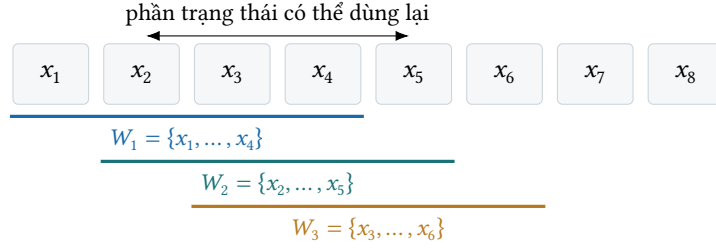
5.3. Phân hoạch thành khối

Chia chuỗi thành các khối liên tiếp

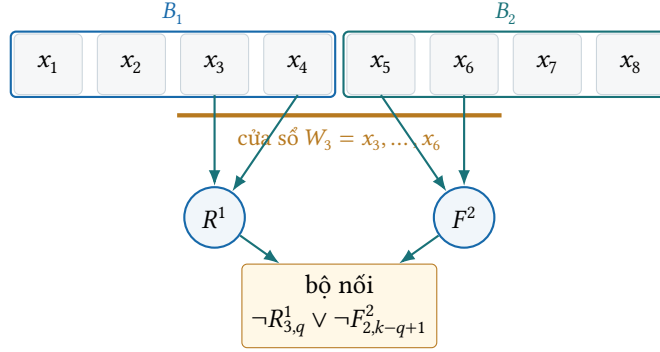
$$B_1, B_2, \dots, B_m, \quad \Omega = B_1 \cdot B_2 \cdots B_m.$$

Trong mỗi khối, xây hai họ biến ngưỡng:

$$F_{i,q}^b = 1 \iff \sum_{j=1}^i B_b[j] \geq q,$$



Hình 5.2.: Ba cửa sổ độ rộng 4 chỉ dịch một vị trí: cửa sổ kế nhau giữ lại ba trong bốn biến. Xây lại một bộ đếm cho từng hàng sẽ lặp phần giao này.



Hình 5.3.: Một cửa sổ giao biên được phân rã thành hậu tố của khối trái và tiền tố của khối phải; bộ nối chỉ đọc các ngưỡng cần thiết.

$$R_{i,q}^b = 1 \iff \sum_{j=i}^{|B_b|} B_b[j] \geq q.$$

F đếm tiền tố từ trái sang phải; R đếm hậu tố từ phải sang trái. Một cửa sổ giao biên giữa B_b, B_{b+1} là hợp của một hậu tố bên trái và một tiền tố bên phải. Số literal đúng trong cửa sổ được đọc từ R^b và F^{b+1} .

Ví dụ 5.1. Giả sử cửa sổ gồm hậu tố A của khối trái và tiền tố B của khối phải. Đặt $T(S, q)$ là literal “ $\sum_{x \in S} x \geq q$ ”, với hai quy ước biên $T(S, q) = \top$ khi $q \leq 0$ và $T(S, q) = \perp$ khi $q > |S|$. Điều kiện at-most- k tương đương với điều kiện không tồn tại $q \in \{0, \dots, k+1\}$ sao cho

$$\sum_{x \in A} x \geq q \quad \text{và} \quad \sum_{x \in B} x \geq k+1-q.$$

Bộ nối vì thế chứa

$$\neg T(A, q) \vee \neg T(B, k+1-q), \quad q = 0, \dots, k+1.$$

Hai giá trị biên $q = 0$ và $q = k+1$ không phải chi tiết hình thức: chúng phát hiện trường hợp toàn bộ $k+1$ giá trị đúng nằm ở chỉ một phía của biên.

Lấy $n = 8, w = 4, k = 2$, chia $B_1 = (x_1, \dots, x_4)$, $B_2 = (x_5, \dots, x_8)$, và xét

$$x = (1, 0, 1, 0 \mid 1, 0, 0, 1).$$

Cửa sổ $W_3 = (x_3, x_4 \mid x_5, x_6) = (1, 0 \mid 1, 0)$ có tổng 2. Hậu tố trái đạt ngưỡng 1 nhưng không đạt 2: $R_{3,1}^1 = 1, R_{3,2}^1 = 0$. Tiền tố phải cũng có $F_{2,1}^2 = 1, F_{2,2}^2 = 0$. Với $k = 2$, bốn bộ nối ứng với $q = 0, 1, 2, 3$ cấm các cặp ngưỡng $(\top, F_3), (R_1, F_2), (R_2, F_1), (R_3, \top)$. Tất cả đều thỏa: hai mệnh đề biên xác nhận mỗi phía riêng lẻ không đạt 3, còn ở hai mệnh đề giữa luôn có một ngưỡng bằng 0.

Nếu đổi x_6 thành 1, tiền tố phải đạt ngưỡng 2. Khi đó $\neg R_{3,1}^1 \vee \neg F_{2,2}^2$ sai, phát hiện ngay tổng của sổ bằng 3. Ví dụ cho thấy bộ nối không cộng lại toàn của sổ; mô-đun chỉ ghép hai bằng chúng ngưỡng.

5.4. Bộ nối như một giao diện

Bộ đếm cục bộ không phụ thuộc vào cửa sổ tiêu thụ; mô-đun chỉ cung cấp đặc tả của các literal ngưỡng. Bộ nối biết cách một ràng buộc giao biên và chọn các ngưỡng cần ghép.

Với at-least- k :

$$\sum_{x \in A \cup B} x \geq k \iff \bigwedge_{q=1}^k \left[\left(\sum_{x \in A} x \geq q \right) \vee \left(\sum_{x \in B} x \geq k - q + 1 \right) \right].$$

Nếu thanh ghi có đặc tả tương đương, mỗi ngoặc vuông trở thành

$$R_{a,q}^b \vee F_{c,k-q+1}^{b+1}.$$

Bộ đếm là bên cung cấp literal ngưỡng; bộ nối là bên sử dụng. Đặc tả hai chiều cần thiết khi bộ nối dùng cả thanh ghi dương và phủ định để diễn giải “đạt” và “chưa đạt” ngưỡng.

5.5. Staircase AMO và SCL

Với $k = 1$, mỗi cửa sổ thỏa AMO. *Shared Counter Ladder* (thang Shared Counter, SCL) xây các trạng thái AMO/AMZ trong khối và nối chúng qua các biên. So với phương án tạo xung đột từng cặp trong từng cửa sổ, SCL không lập các cặp nằm trong phần giao.

Giả sử $m = \lceil n/w \rceil$ khối có kích thước gần w . Chi phí bộ đếm cục bộ tuyến tính theo tổng kích thước khối, còn các bộ nối tuyến tính theo số biên và ngưỡng. Vì vậy kích thước toàn bộ có thể giảm từ bậc hai về tuyến tính theo n ở các AMO bậc thang điển hình.

Biểu diễn Duplex dùng BDD tiến/lùi trong mỗi cửa sổ và liên kết các trạng thái [35]. SCL dùng trạng thái tuần tự, nhờ đó giảm số biến phụ và mệnh đề trong phân tích của Truong, Kieu, và To [80].

Trong hệ phân loại trên, SCL phục vụ các tập xung đột có dạng khoảng dịch chuyển và phần giao lớn giữa các khoảng kề nhau. Biểu diễn từng cặp và bộ đếm tuần tự độc lập phù hợp hơn với các tập AMO rời rạc hoặc thiếu một thứ tự tuyến tính ổn định. Độ chồng lấn vì thế là đặc trưng đầu tiên để chọn giữa trạng thái dùng chung và bộ đếm cục bộ.

Trên họ SCAMO với $n = 1000$ và độ rộng cửa sổ thay đổi, Truong, Kieu, và To [80] báo SCL dùng ít biến và mệnh đề hơn Duplex cùng các phép biểu diễn từng cửa sổ. Khi đưa vào bài toán Anti-Bandwidth, SCL đạt kết quả cạnh tranh trên 24 đồ thị kiểm chuẩn, đặc biệt ở các bài toán lớn nơi chi phí lập cửa sổ trở nên rõ rệt.

5.6. Mở rộng sang Ladder AMK

Khi $k > 1$, mỗi khối cần nhiều mức ngưỡng. Ta dùng Sequential Counter:

$$F_{i,q}^b \leftrightarrow F_{i-1,q}^b \vee (B_b[i] \wedge F_{i-1,q-1}^b),$$

và hệ thức truy hồi đối xứng cho R^b .

Bộ nối cho cận trên cấm mọi cặp ngưỡng có tổng $k + 1$:

$$\neg T(A, q) \vee \neg T(B, k + 1 - q), \quad q = 0, \dots, k + 1.$$

Vì thế các thanh ghi thực phải được xây đến ngưỡng $k + 1$, trừ những mức có thể giản lược thành $\top$ hoặc $\perp$ theo kích thước phần của sổ. Nếu một cửa sổ đi qua nhiều hơn một biên, có hai lựa chọn:

1. tăng độ rộng khối để mỗi cửa sổ cắt nhiều nhất một biên;
2. xây bộ nối nhiều ngôi hoặc một bộ đếm trung gian trên tổng của các khối.

Thiết kế thứ nhất đơn giản hơn; thiết kế thứ hai linh hoạt với nhiều width.

Ladder AMK trong Truong, Kieu, và To [81] phát triển Sequential Counter dùng chung cho một tập ràng buộc đếm dạng bậc thang. Giá trị cốt lõi không nằm ở một mệnh đề riêng lẻ mà ở việc bộ đếm được tạo cho cả họ ràng buộc.

5.6.1. Quan hệ với tổng tiền tố và ràng buộc chuỗi tổng quát

Ladder AMK có thể được mô tả bằng ma trận liên thuộc: mỗi hàng là một ràng buộc, mỗi cột là một đầu vào, và các giá trị 1 trong mỗi hàng tạo thành một khoảng; các biên khoảng thay đổi đơn điệu theo hàng. Biểu diễn tổng tiền tố dùng một trạng thái toàn cục cho mọi cột. Ladder AMK chỉ hiện thực hóa (reification) những ngưỡng tiền tố/hậu tố mà các hàng thật sự đọc. Do đó tiêu chí lựa chọn không chỉ là n, k , mà còn là độ phức tạp của hai đường biên bậc thang.

Với nhiều độ rộng và nhiều cận, một ô tô-mát có thể nắm toàn bộ lịch sử nhưng số trạng thái tăng theo tổ hợp các bộ đếm đang hoạt động. Số ngưỡng chia sẻ tăng theo số biên và mức đếm được yêu cầu. Hai hướng vì thế tạo một đánh đổi: ô tô-mát hợp nhất các ràng buộc trong trạng thái, còn Ladder AMK giữ các ràng buộc tách biệt nhưng chia sẻ đại lượng trung gian. Đây là câu hỏi thực nghiệm và lý thuyết mang nhiều ý nghĩa hơn so với việc chỉ so sánh số lượng mệnh đề trên một cấu hình.

5.7. Đúng đắn

Định lý 5.2 (Tính đúng của phép biểu diễn cửa sổ chia sẻ). *Giả sử:*

1. các thanh ghi tiến/lùi đúng theo đặc tả ngưỡng;
2. mọi cửa sổ nằm trong một khối được ràng buộc bởi bộ đếm cục bộ;
3. mỗi cửa sổ hoặc nằm trong một khối, hoặc cắt đúng một biên khối;
4. với cửa sổ $A \cup B$ giao biên, phép biểu diễn chứa đủ các mệnh đề

$$\neg T(A, q) \vee \neg T(B, k + 1 - q), \quad q = 0, \dots, k + 1.$$

Khi đó CNF tương đương theo phép chiếu với họ staircase AMK.

Chứng minh. Để chứng minh tính đúng, xét một cửa sổ. Nếu nằm trong khối, bộ đếm cục bộ loại mọi tổng vượt k . Nếu giao biên, đặt $a = \sum_{x \in A} x$, $b = \sum_{x \in B} x$. Khi $a + b \geq k + 1$, chọn $q = \min\{a, k + 1\}$. Ta có $T(A, q) = 1$; nếu $q < k + 1$ thì $b \geq k + 1 - a = k + 1 - q$, còn nếu $q = k + 1$ thì ngưỡng bên phải bằng 0 và đúng theo quy ước. Bộ nối tương ứng bị vi phạm.

Để chứng minh tính đầy đủ, lấy một chuỗi thỏa mọi cửa sổ và gán các thanh ghi bằng đúng tổng tiền tố/hậu tố. Các mệnh đề của bộ đếm đều đúng. Nếu một bộ nối sai thì tồn tại q sao cho $a \geq q$ và $b \geq k + 1 - q$, suy ra $a + b \geq k + 1$, mâu thuẫn với giả thiết cửa sổ nhiều nhất k . Do đó phép gán trên biến gốc luôn mở rộng được thành mô hình của CNF; kết hợp với tính đúng cho tương đương theo phép chiếu. $\square$

5.7.1. Cửa sổ cắt nhiều biên

Định lý 5.2 bao phủ trực tiếp cửa sổ trong một khối và cửa sổ cắt một biên. Với cửa sổ đi qua ba khối trở lên, tổng được ghép từ nhiều hơn hai thành phần. Có thể ghép tuần tự các khối bằng một bộ đếm trung gian, hoặc dùng một cây ghép có đầu ra bị cắt ở $k + 1$. Với các tổng bộ phận $S_1, \dots, S_r$, bất biến cần giữ là

$$U_{j,q} \iff \min\left(k + 1, \sum_{h=1}^j S_h\right) \geq q.$$

Sau lần ghép cuối, mệnh đề $\neg U_{r,k+1}$ đóng AMK. Chi phí tăng theo số khối mà từng cửa sổ chạm tới; một cây ghép riêng cho mỗi cửa sổ làm giảm tỷ lệ trạng thái chia sẻ. Điều kiện “mỗi cửa sổ cắt nhiều nhất một biên” do đó là tiêu chí phân hoạch, còn bộ đếm ghép là phương án cho phân hoạch tổng quát.

5.8. Chọn độ rộng khối

Khối nhỏ tạo nhiều biên và bộ nối nhưng bộ đếm cục bộ ngắn. Khối lớn giảm số biên nhưng giữ nhiều thanh ghi không được mọi cửa sổ dùng. Vì vậy độ rộng là một siêu tham số cấu trúc.

Ba cách chọn:

- bằng độ rộng cửa sổ khi mọi ràng buộc có cùng w ;
- theo ước chung lớn nhất hoặc cụm của nhiều độ rộng;
- phân hoạch thích nghi, đặt biên tại nơi có ít cửa sổ cắt qua.

Phân hoạch thích nghi có thể xem là bài toán tối ưu trên đồ thị khoảng: mỗi cửa sổ là một khoảng, chi phí của một biên bằng số ràng buộc giao biên nhân với số ngưỡng cần nối.

5.9. Triển khai

Application programming interface (giao diện lập trình, API) cho mô-đun có thể là:

SharedCounter(sequence, partition, maxThreshold)

và trả về:

- các literal ngưỡng tiền tố/hậu tố;
- hàm `prefix(block, pos, q)`;
- hàm `suffix(block, pos, q)`;
- ánh xạ thanh ghi để gỡ lỗi và kiểm tra nội bộ.

Sau đó `addAtMostWindow(i, w, k)` chỉ sinh các bộ nối. Nhiều mô-đun sử dụng có thể đọc cùng một bộ đếm: AMK, ALK, EXK hoặc hàm mục tiêu.

Đầu vào: chuỗi $x_1, \dots, x_n$; một phân hoạch khối; tập cửa sổ $\mathcal{W}$; cận $k(W)$ của từng cửa sổ.

Đầu ra: một CNF tương đương theo phép chiếu.

1. Trong mỗi khối, sinh các thanh ghi ngưỡng tiến/lùi đến mức lớn nhất $\max_{W \in \mathcal{W}}(k(W) + 1)$ thật sự được đọc.
2. Với cửa sổ nằm trọn trong một khối, đọc bộ đếm cục bộ và cấm ngưỡng $k(W) + 1$.
3. Với cửa sổ $W = A \cup B$ cắt một biên, lần lượt $q = 0, \dots, k(W) + 1$, sinh $\neg T(A, q) \vee \neg T(B, k(W) + 1 - q)$; giản lược các literal hằng.
4. Nếu phát hiện một cửa sổ cắt từ hai biên trở lên, dừng mô-đun này và chuyển sang bộ đếm ghép; không áp dụng bộ nối hai ngôi.
5. Trả CNF cùng ánh xạ từ mỗi thanh ghi về (khối, vị trí, q).

Thuật toán 5.1: Sinh AMK của sổ chia sẻ khi mỗi cửa sổ cắt nhiều nhất một biên

5.9.1. Miền hiệu quả của Shared Counter

Shared Counter đạt lợi thế rõ nhất khi các cửa sổ chồng lấn mạnh, cận k nhỏ và mỗi cửa sổ chỉ cắt ít khối. Với các cửa sổ gần rời nhau hoặc bài toán nhỏ, biểu diễn từng cặp và bộ đếm độc lập thường có hằng số thấp hơn. Một bộ chọn có thể quyết định trước khi sinh CNF từ ba đại lượng: tổng số ngưỡng cục bộ, số bộ nối sau giản lược và số bước ghép của các cửa sổ nhiều biên.

Ngoài các ràng buộc cửa sổ trượt SEQUENCE, nguyên lý tái sử dụng trạng thái phụ qua các điểm cắt miền còn được mở rộng thành kỹ thuật hậu tố thừa SPARSE SUFFIX trong bài toán lập lịch cuộc họp doanh nghiệp (B2B Meeting Scheduling) [64] (xem Mục 8.5). Tại đó, các biến đại diện hậu tố được chia sẻ trực tiếp giữa các quan hệ thứ tự tiên quyết có cùng điểm cắt miền tiên nhiệm, giúp giảm đáng kể số mệnh đề nhị phân sinh ra trong công thức CNF.

5.10. Bài học thiết kế

1. Cửa sổ chồng lấn là tín hiệu nên dùng trạng thái chung.
2. Bộ đếm cục bộ cung cấp các ngưỡng; bộ nối mô tả ràng buộc.
3. Thanh ghi tiến và lùi tách một cửa sổ giao biên thành hai phần.
4. Chứng minh tính đúng tách cửa sổ nội khối khỏi cửa sổ giao biên.
5. Độ rộng khối cân bằng chi phí thanh ghi với chi phí bộ nối.

Câu hỏi nghiên cứu

1. Dựng các bộ nối cho một cửa sổ AMK với $k = 2$ giao biên thành hai phần.
2. Tính số thanh ghi khi $n = 20, w = 5, k = 3$ và độ rộng khối bằng 5.
3. Đề xuất hàm mục tiêu cho phân hoạch thích nghi dựa trên số cửa sổ giao biên.

6 Thanh ghi đếm thích nghi cho ràng buộc đúng K

TÓM TẮT CHƯƠNG

Sequential Counter thông thường dành cùng một độ rộng cho mọi hàng. Thanh ghi thích nghi chỉ tạo những trạng thái đếm có thể đạt tới, nhờ đó giữ được cấu trúc tuần tự nhưng loại bỏ phần tam giác trạng thái bất khả.

6.1. Ràng buộc đúng K trong không gian thiết kế

Exactly- K (ràng buộc đúng K) có nhiều cách phân rã đúng:

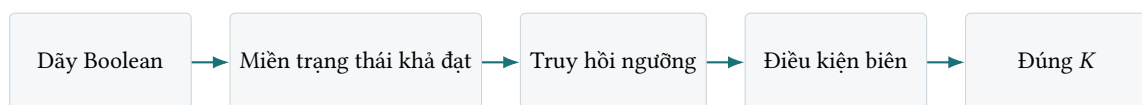
$$\sum X = K \iff (\sum X \leq K) \wedge (\sum X \geq K),$$

nhưng hai vế không nhất thiết phải dùng hai bộ biểu diễn độc lập. Một vectơ đơn phân đầy đủ cho phép đọc cùng lúc o_K và $\neg o_{K+1}$; BDD giữ các trạng thái tổng; Sorting Network các đầu vào; Sequential Counter theo dõi tiền tố. Câu hỏi của chương là cụ thể: nếu đã chọn các trạng thái ngưỡng tuần tự thì những ô nào thật sự có thể đạt và cần được hiện thực hóa (reification)?

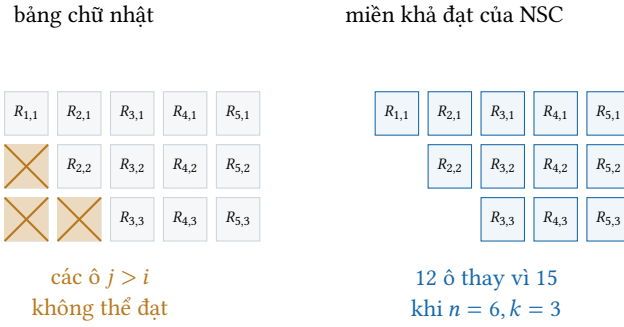
Bảng 6.1.: Các hướng biểu diễn ràng buộc đúng K .

Họ	Trạng thái	Giao diện EXK	Đặc trưng
Hai AMK	Hai bộ đếm	Hai cận tách	Dễ ghép nhưng có thể lặp trạng thái
Tuần tự	Tổng tiền tố	Chống tràn + đạt K	Tốt khi K nhỏ
Totalizer	Tổng cây con	$o_K \wedge \neg o_{K+1}$	Dùng lại nhiều cận
Mạng	Vị trí đã sắp	Hai đầu ra kề nhau	Cấu trúc đều, độ sâu thấp
BDD	(i, s)	Chấp nhận trạng thái K	Hợp nhất trạng thái theo tổng
NSC	Ngưỡng tiền tố khả đạt	Biên trên + dưới	Cắt vùng tam giác bất khả

Các cận dưới cho AMO và những khảo cứu về ràng buộc đếm đặt số biến bên cạnh số mệnh đề, literal và mức lan truyền [8, 54]. NSC giữ bậc $O(nK)$, giảm một phần xác định của bảng và thay cách đóng biên Exactly- K . Đóng góp cốt lõi là một *không gian trạng thái gọn hơn trong họ tuần tự*; vị trí trên biên Pareto (Pareto frontier) được đo bằng số mệnh đề, số literal và sức lan truyền.



Hình 6.1.: Lộ trình lập luận của chương: từ Dây Boolean, qua Miền trạng thái khả đạt, Truy hồi ngưỡng và Điều kiện biên, đến Đúng K .



Hình 6.2.: Sequential Counter hình chữ nhật và miền trạng thái tam giác. NSC không tạo ba ô bất khả $R_{1,2}, R_{1,3}, R_{2,3}$.

6.2. Từ bảng chữ nhật đến miền trạng thái khả đạt

Xét dãy Boolean $X = \langle x_1, \dots, x_n \rangle$ và ràng buộc

$$\sum_{i=1}^n x_i = k.$$

Một Sequential Counter có thể dành k cột cho mọi tiền tố. Tuy nhiên, sau i biến đầu tiên, tổng không thể lớn hơn i . Các ô $j > i$ vì thế không mang trạng thái hợp lệ.

New Sequential Counter (Sequential Counter mới, NSC) dùng thanh ghi tam giác

$$R_{i,j} = 1 \iff \sum_{q=1}^i x_q \geq j, \quad \begin{matrix} 1 \leq i \leq n-1, \\ 1 \leq j \leq \min(i, k). \end{matrix}$$

Chỉ $n-1$ tiền tố được hiện thực hóa (reification); biến x_n được xử lý trong khi giải (in-processing) khi giải (in-processing) điều kiện cuối. Với $n > k$, số biến phụ là

$$\sum_{i=1}^{n-1} \min(i, k) = k(n-1) - \frac{k(k-1)}{2}.$$

So với bảng chữ nhật $k(n-1)$, phần tiết kiệm chính xác là $k(k-1)/2$ biến. Bậc tiệm cận được giữ nguyên, còn hệ số hữu hạn giảm đúng các trạng thái bất khả mà bộ giải phải duy trì.

Độ rộng hàng i là $\min(i, k)$, được quyết định bởi số trạng thái có thể đạt sau i đầu vào. “Thích nghi” ở đây mô tả hình dạng tĩnh của bộ đếm, được xác định trước khi bộ giải chạy.

6.3. Đặc tả ngữ nghĩa và hệ thức truy hồi

Dùng hai quy ước biên

$$R_{i,0} \equiv \top, \quad R_{0,j} \equiv \perp \quad (j \geq 1).$$

Khi đó mọi ô hợp lệ tuân theo

$$R_{i,j} \iff R_{i-1,j} \vee (x_i \wedge R_{i-1,j-1}). \quad (6.1)$$

Về phải mô tả đúng hai cách để tiền tố mới đạt ngưỡng j : tiền tố cũ đã đạt ngưỡng, hoặc $x_i = 1$ và tiền tố cũ đã đạt $j-1$.

Trong CNF, hệ thức truy hồi được phân rã thành các quan hệ kéo theo theo hai chiều. Phần tiến gồm

$$R_{i-1,j} \rightarrow R_{i,j}, \quad (x_i \wedge R_{i-1,j-1}) \rightarrow R_{i,j}.$$

Phần lùi gồm

$$(\neg x_i \wedge \neg R_{i-1,j}) \rightarrow \neg R_{i,j}, \quad \neg R_{i-1,j-1} \rightarrow \neg R_{i,j}.$$

Sau khi thay các hằng biên và bỏ literal thừa, ta thu được các mệnh đề hai hoặc ba literal. Cách sinh mệnh đề trực tiếp theo miền tam giác tránh tạo biến rồi giao cho bộ tiền xử lý xóa đi.

Nếu $R_{i,j}$ xuất hiện trong điều kiện biên, hàm mục tiêu hoặc một ràng buộc khác, hệ thức truy hồi phải bảo đảm đặc tả hai chiều. Chỉ các quan hệ kéo theo tiến không đủ để diễn giải $\neg R_{i,j}$ là “tiền tố chưa đạt j ”.

6.4. Đóng ràng buộc đúng K

Phần nhiều nhất k cấm một đầu vào mới làm tràn bộ đếm:

$$x_i \rightarrow \neg R_{i-1,k}, \quad i = k + 1, \dots, n. \quad (6.2)$$

Phần ít nhất k yêu cầu tổng cuối đạt ngưỡng k :

$$R_{n-1,k} \vee (x_n \wedge R_{n-1,k-1}). \quad (6.3)$$

Biểu thức công thức (6.3) được phân phối thành CNF trước khi xuất. Khi $k = 1$, $R_{n-1,0} \equiv \top$; khi $k = n$, có thể đơn giản hóa toàn bộ ràng buộc thành các mệnh đề đơn vị x_i .

Định lý 6.1 (Đúng đắn của NSC). *Hệ thức truy hồi công thức (6.1) cùng các điều kiện các công thức (6.2) and (6.3) tương đương theo phép chiếu với $\sum_{i=1}^n x_i = k$.*

Chứng minh. Hệ thức truy hồi và các hằng biên cho, bằng quy nạp theo i ,

$$R_{i,j} = 1 \iff \sum_{q=1}^i x_q \geq j.$$

Nếu có hơn k đầu vào đúng, xét đầu vào đúng thứ $k + 1$, ký hiệu x_i . Ngay tại vị trí trước đó, $R_{i-1,k} = 1$, trái với công thức (6.2). Do đó tổng không vượt k .

Ngược lại, công thức (6.3) chỉ đúng khi k đầu vào đầu tiên đã xuất hiện trong $n - 1$ vị trí đầu, hoặc $x_n = 1$ và tiền tố trước đó chứa ít nhất $k - 1$ đầu vào đúng. Vì vậy tổng ít nhất k . Hai cận cho tổng đúng bằng k . Với mọi vectơ X có tổng k , gán $R_{i,j}$ theo tổng tiền tố thật thỏa hệ thức truy hồi và cả hai điều kiện biên. □

Với $n = 6, k = 3$, NSC tạo

$$\min(1, 3) + \min(2, 3) + \dots + \min(5, 3) = 12$$

biến thành ghi, thay vì $5 \cdot 3 = 15$ ô của bảng chữ nhật. Xét

$$x = (1, 0, 1, 0, 0, 1).$$

Sau năm biến, tổng tiền tố bằng 2, nên

$$(R_{5,1}, R_{5,2}, R_{5,3}) = (1, 1, 0).$$

Điều kiện cuối $R_{5,3} \vee (x_6 \wedge R_{5,2})$ đúng chính vì $x_6 = 1$ và tiền tố đã đạt ngưỡng 2. Nếu đặt thêm $x_5 = 1$, ta có $R_{5,3} = 1$; mệnh đề chống tràn $x_6 \rightarrow \neg R_{5,3}$ lập tức buộc $x_6 = 0$. Hai tình huống này minh họa lần lượt biên dưới và biên trên của cùng một thanh ghi.

6.5. Cơ chế suy diễn lan truyền hai chiều

Bộ đếm hữu ích vì các cận xuất hiện sớm trong tìm kiếm. Hai tình huống quan trọng có thể kiểm tra trực tiếp:

1. Khi đã có k đầu vào đúng trong một tiền tố, thanh ghi mức k của tiền tố đó trở thành đúng. Các mệnh đề chống tràn buộc mọi đầu vào còn lại về 0.
2. Khi số vị trí có thể còn nhận giá trị 1 vừa đủ để đạt k , điều kiện cận dưới và các quan hệ kéo theo lùi lần lượt loại nhánh đặt một vị trí còn lại về 0.

Hai hướng lan truyền này tương ứng với hai biên “đủ” và “phải dùng hết phần còn lại”. Chúng nên được kiểm tra bằng phép thử đơn vị trên mọi phép gán bộ phận nhỏ, thay vì suy luận hiệu quả chỉ từ số biến hoặc số mệnh đề.

6.6. Dùng một thanh ghi cho nhiều giao diện

Khi thanh ghi đã có đặc tả ngưỡng, nhiều ràng buộc có thể đọc cùng một mô-đun:

$$\sum X \leq k \implies \text{các mệnh đề chống tràn,}$$

$$\sum X \geq k \implies \text{đầu ra đạt ngưỡng cuối,}$$

$$\sum X = k \implies \text{kết hợp hai giao diện trên.}$$

Nếu cần nhiều cận $k_1 < k_2 < \dots$, bộ đếm nên được tạo đến ngưỡng lớn nhất rồi cung cấp các đầu ra nhỏ hơn. Cách này tránh xây hai mạng đếm độc lập cho ràng buộc và hàm mục tiêu.

6.7. Nghiên cứu tình huống ngắn: xếp nhóm chơi golf

6.7.1. Từ thiết kế tổ hợp đến mô hình SAT

Social Golfer (bài toán xếp nhóm người chơi golf) vừa là bài toán lập lịch vừa là bài toán thiết kế tổ hợp. Ba nguồn khó đan xen với nhau: ràng buộc đúng kích thước nhóm, cấm lặp cặp người chơi và một nhóm đối xứng lớn do hoán vị tuần, nhóm, người. Vì vậy, hiệu quả của một phép biểu diễn ràng buộc đếm trên SGP không thể tách khỏi lựa chọn biến chính và cách phá vỡ đối xứng.

Các mô hình SAT trước đó từng dùng thêm chỉ số vị trí trong nhóm, sau đó được sửa và rút gọn bằng biến người–nhóm–tuần cùng cấu trúc bậc thang [79]. Dòng phát triển này minh họa một quy tắc quan trọng: bỏ một chỉ số không mang nghĩa thường tạo lợi ích lớn hơn tối ưu vài mệnh đề trong chính bộ biểu diễn ràng buộc đếm. Phép đánh giá NSC vì thế bắt đầu sau khi phân biểu diễn đối tượng đã được chuẩn hóa.

Một bài toán thử g - p - w có $n = gp$ người chơi, g nhóm mỗi tuần, mỗi nhóm có p người và w tuần. Biến ba chỉ số

$$G_{i,r,t} = 1 \iff \text{người chơi } i \text{ thuộc nhóm } r \text{ ở tuần } t$$

loại bỏ chỉ số vị trí bên trong nhóm, bởi thứ tự các thành viên trong một nhóm không mang ý nghĩa.

tuần	nhóm 1	nhóm 2	nhóm 3
1	{1, 2}	{3, 4}	{5, 6}
2	{1, 3}	{2, 5}	{4, 6}

mỗi người đúng một nhóm/tuần;
mỗi cặp gặp nhiều nhất một lần

Hình 6.3.: Một lịch khả thi cho bài toán nhỏ 3-2-2. Mỗi ô nhóm là một ràng buộc đúng 2; tuần đầu có thể được cố định làm lát chuẩn.

Mô hình gồm ba họ ràng buộc:

$$\sum_{r=1}^g G_{i,r,t} = 1 \quad \text{với mọi người chơi } i, \text{ tuần } t,$$

$$\sum_{i=1}^n G_{i,r,t} = p \quad \text{với mọi nhóm } r, \text{ tuần } t,$$

$$\neg G_{i,r,t} \vee \neg G_{j,r,t} \vee \neg G_{i,s,u} \vee \neg G_{j,s,u} \quad \text{với } i < j, t < u.$$

Họ thứ ba cấm một cặp người chơi gặp nhau ở cả nhóm r , tuần t và nhóm s , tuần u . NSC được dùng ở họ thứ hai; mỗi cặp nhóm–tuần có một bộ đếm tam giác riêng [50].

Hai phép chuẩn hóa rẽ được thêm vào mô hình. Tuần đầu được cố định thành các khối người chơi liên tiếp; ở các tuần sau, một số người chơi neo được gán vào các nhóm đã định. Chúng loại các hoán vị hiển nhiên của người chơi, nhóm và tuần mà không thêm biến phụ.

Trong thí nghiệm của Kieu, Nguyen, và To [50], cấu hình dùng NSC giải được 56/66 bài toán; các cấu hình ADDER, BDD và CARD cùng giải được 54, còn BINOMIAL giải được 32. NSC có số lần hết thời hạn thấp nhất; thứ hạng trên từng bài toán thay đổi theo cấu trúc của công thức. Kết quả làm rõ lợi ích của việc cân bằng số biến, số mệnh đề và cấu trúc lan truyền.

6.7.2. Phân tích kết quả

Mô hình ba chỉ số loại đối xứng vị trí, còn trạng thái tuần tự tam giác cung cấp một phép biểu diễn EXK cạnh tranh cho ràng buộc kích thước nhóm. Có thể đo riêng tác động của NSC bằng cách giữ cố định mô hình SGP, chỉ thay mô-đun kích thước nhóm, phân tầng theo p , g , w và tách các bài SAT/UNSAT.

6.8. Cài đặt và kiểm thử

Một bộ sinh NSC nên nhận dãy literal, cận k và chế độ AMK/ALK/EXK. Bảng ánh xạ $(i, j) \mapsto \text{varID}$ chỉ chứa ô hợp lệ. Ba lớp kiểm thử là đủ để bắt phần lớn lỗi biên:

- liệt kê toàn bộ vectơ đầu vào khi $n \leq 8$ và so sánh với tổng thật;
- với mỗi vectơ hợp lệ, kiểm tra có tồn tại phần mở rộng cho biến R ;
- với mỗi phép gán một phần, so sánh literal đầu vào mà lan truyền đơn vị suy ra với các hệ quả trực tiếp của hai cận.

Các trường hợp $k = 0$, $k = 1$, $k = n - 1$, $k = n$ cần được kiểm thử riêng. Đây là nơi quy ước hằng biên và phạm vi chỉ số thường gây lỗi nhất.

Đầu vào: các literal $x_1, \dots, x_n$ và $0 \leq k \leq n$.

Đầu ra: CNF và ánh xạ ngữ nghĩa của các thanh ghi.

1. Xử lý trực tiếp $k = 0$ bằng $\bigwedge_i \neg x_i$ và $k = n$ bằng $\bigwedge_i x_i$. Nếu $k > n/2$, có thể thay $y_i = \neg x_i$ và đếm Exactly- $(n - k)$.
2. Chỉ cấp phát $R_{i,j}$ với $1 \leq i \leq n - 1$ và $1 \leq j \leq \min(i, k)$; ghi ánh xạ $R_{i,j} \leftrightarrow [\sum_{q=1}^i x_q \geq j]$.
3. Với từng ô hợp lệ, phân rã tương đương $R_{i,j} \leftrightarrow R_{i-1,j} \vee (x_i \wedge R_{i-1,j-1})$, thay $R_{i,0} = \top$, $R_{0,j} = \perp$, rồi giản lược các literal hằng.
4. Sinh các mệnh đề chống tràn của (6.2).
5. Sinh điều kiện đạt cận dưới (6.3); trả CNF và ánh xạ.

Thuật toán 6.1: Sinh NSC cho ràng buộc đúng K

6.9. Miền lựa chọn của NSC

NSC loại chính xác $k(k - 1)/2$ ô so với bảng chữ nhật, nên tỷ lệ tiết kiệm biến chỉ là

$$\frac{k(k - 1)/2}{k(n - 1)} = \frac{k - 1}{2(n - 1)}.$$

Khi n lớn nhưng k cố định, tỷ lệ này tiến về 0; hằng số của ánh xạ và các mệnh đề hai chiều có thể quan trọng hơn phần tam giác bị bỏ. Với k gần n , đếm các literal bù $\neg x_i$ đến $n - k$ thường gọn hơn. Nếu cùng một tập đầu vào cần nhiều cận hoặc cần thay cận tăng dần, Totalizer và mạng đếm có đầu ra chia sẻ tự nhiên hơn một NSC chuyên biệt. BDD cũng có thể thắng khi phép rút gọn hợp nhất được nhiều trạng thái tương đương.

Bộ chọn cho ràng buộc đúng K dùng n , $\min(k, n - k)$, số cận sẽ được truy vấn, nhu cầu giải tăng dần và đặc tả lan truyền làm đặc trưng. NSC phù hợp với một cận cố định và vùng tam giác bị cắt đủ lớn; Totalizer/mạng đếm phù hợp khi cần chia sẻ nhiều đầu ra. Kết quả Social Golfer minh họa cách các đặc trưng này dẫn đến một lựa chọn hiệu quả trên họ bài toán tương ứng.

6.10. Bài học thiết kế

1. Miền trạng thái khả đạt quyết định hình dạng bộ đếm.
2. Thanh ghi $R_{i,j}$ có đặc tả ngưỡng rõ ràng, không chỉ là biến Tseitin vô danh.
3. Ràng buộc đúng K là sự kết hợp của một biên tràn và một điều kiện đạt ngưỡng cuối.
4. Hiệu quả cần được đọc đồng thời qua kích thước, lan truyền và hành vi giải trên bài toán khó.
5. Social Golfer cho thấy lựa chọn biến chính và phép biểu diễn ràng buộc đếm phải được thiết kế cùng nhau.

Câu hỏi nghiên cứu

1. Vẽ miền biến $R_{i,j}$ cho $n = 7, k = 3$ và tính phần tiết kiệm so với bảng chữ nhật.
2. Viết các mệnh đề chống tràn cho $\sum_{i=1}^6 x_i \leq 2$.
3. So sánh hai mô hình Social Golfer có và không có chỉ số vị trí trong nhóm về số biến chính và nhóm đối xứng còn lại.

7 Đối xứng, biến quan hệ và liên kết biểu diễn

TÓM TẮT CHƯƠNG

Một phép biểu diễn tốt không chỉ mô tả nghiệm; mô hình còn quyết định cách bộ giải nhìn thấy các nghiệm tương đương và các xung đột cục bộ. Symmetry breaking (phá vỡ đối xứng) thu gọn không gian tìm kiếm, còn channeling (liên kết biểu diễn) nối các tiêu chí hỗ trợ của cùng một quyết định.

7.1. Hai cấp đối xứng: mô hình và công thức

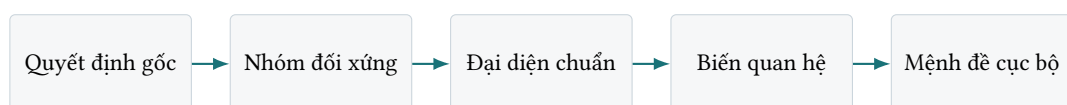
Một đối xứng của bài toán tác động lên đối tượng miền: đổi tên khung chứa, máy, nhóm, tuần, màu hoặc phản xạ hình học. Một đối xứng của CNF là hoán vị biến và literal bảo toàn tập mệnh đề. Hai nhóm này liên quan nhưng không trùng nhau. Phép biểu diễn có thể:

- làm một đối xứng miền biến mất vì không biểu diễn chỉ số thừa;
- làm một đối xứng miền trở nên khó phát hiện do biến phụ;
- tạo đối xứng phụ mới giữa các trạng thái tương đương;
- có đối xứng cú pháp không tương ứng với một phép biến đổi nghiệm dễ diễn giải.

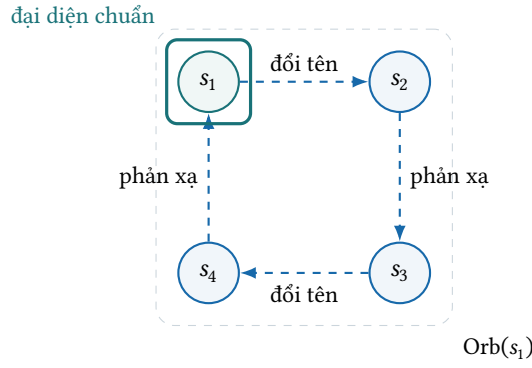
Các vị từ phá vỡ đối xứng tổng quát xây một điều kiện chọn đại diện cho các quỹ đạo của bài toán tìm kiếm [27]. Ở cấp CNF, những hệ như Shatter xây đồ thị tô màu của công thức, tìm tự đẳng cấu rồi sinh các ràng buộc phá vỡ đối xứng [3]. Ở cấp mô hình, người thiết kế đã biết ngữ nghĩa nên có thể viết ràng buộc ngắn hơn, chẳng hạn tiền tố khung chứa đã dùng. Hai cách bổ sung nhau: phá vỡ đối xứng miền được áp dụng trước; phát hiện trên CNF tìm phần đối xứng còn sót.

Bảng 7.1.: Bốn cấp khai thác đối xứng.

Cấp	Đầu vào	Lợi thế	Hạn chế
Mô hình	Đối tượng và phép đổi tên	Ràng buộc ngắn, có chứng minh miền	Cần tri thức bài toán
CNF tĩnh	Đồ thị công thức	Tự động, dùng cho mô hình bất kỳ	Biến phụ có thể che đối xứng
Động	Trạng thái tìm kiếm	Khai thác đối xứng còn lại	Chi phí tích hợp bộ giải
Chuẩn hóa	Toàn quỹ đạo	Có thể giữ một đại diện	Ràng buộc lớn/lan truyền sâu



Hình 7.1.: Lộ trình lập luận của chương: từ Quyết định gốc, qua Nhóm đối xứng, Đại diện chuẩn và Biến quan hệ, đến Mệnh đề cục bộ.



Hình 7.2.: Một quỹ đạo gồm bốn cách biểu diễn tương đương. Ràng buộc phá đối xứng hợp lệ giữ ít nhất một nút; phép chuẩn hóa mạnh hơn sẽ cố giữ đúng một nút.

Các lớp ràng buộc phá vỡ đối xứng được ghép theo thứ tự. Ràng buộc miền làm thay đổi nhóm tự đẳng cấu của đồ thị; bộ phá vỡ đối xứng tự động vì thể chạy trên công thức đã có mệnh đề đơn vị và liên kết biểu diễn. Mỗi lớp bảo toàn ít nhất một đại diện của các quỹ đạo còn lại, còn thí nghiệm phân tích thành phần (ablation study) đo tác động của từng lớp lên hiệu năng [38].

7.2. Đối xứng nhu tác động lên nghiệm

Gọi S là tập nghiệm khả thi và Γ là một nhóm các phép biến đổi bảo toàn tính khả thi và giá trị mục tiêu. Hai nghiệm $s, s' \in S$ tương đương khi tồn tại $\gamma \in \Gamma$ sao cho $s' = \gamma(s)$. Tập

$$\text{Orb}(s) = \{\gamma(s) \mid \gamma \in \Gamma\}$$

là quỹ đạo của s .

Phá vỡ đối xứng thêm công thức B sao cho mỗi quỹ đạo còn ít nhất một đại diện:

$$\forall s \in S, \quad \text{Orb}(s) \cap \text{Mod}(B) \neq \emptyset.$$

Không nhất thiết phải giữ đúng một đại diện. Trong thực hành, một tập ràng buộc phá vỡ đối xứng nhỏ, rẻ và phù hợp với mô hình thường hữu ích hơn việc cố gắng biểu diễn một dạng chuẩn hoàn chỉnh [38, 88].

Chúng minh một ràng buộc phá vỡ đối xứng bằng phép hoán vị hoặc phép phản xạ đưa mọi nghiệm bị loại sang một nghiệm giữ lại. Ánh xạ này đồng thời xác định nhóm đối xứng và đại diện được bảo toàn.

7.3. Loại đối xứng ngay từ lựa chọn biến

Biến chính có thể tạo hoặc loại đối xứng. Trong Social Golfer, biến $G_{i,r,t}$ chỉ ghi người chơi, nhóm và tuần. Nếu thêm chỉ số vị trí trong nhóm, mỗi nhóm p người tạo thêm $p!$ cách biểu diễn cùng một tập. Trong bài toán đóng gói (packing problem), biến chiếm chỗ theo từng ô có thể tạo nhiều cách diễn giải một hình chữ nhật; biến tọa độ và hướng đặt biểu diễn hình học trực tiếp hơn.

Nguyên tắc chung là:

không Order Encoding nếu thứ tự không thuộc nghiệm.

Sau bước chọn biến, các ràng buộc phá vỡ đối xứng xử lý nhóm đối xứng còn lại. Cách tiếp cận đó giảm cả số biến lẫn số mệnh đề; bước phá vỡ đối xứng tiếp tục giảm số mô hình.

7.4. Bốn mẫu phá vỡ đối xứng

7.4.1. Cố định một lát chuẩn

Khi các lớp thời gian hoặc tài nguyên hoán vị tự do, có thể cố định lát đầu. Với Social Golfer, tuần đầu được chuẩn hóa:

$$G_{i,r,1} = 1 \quad \text{khi} \quad (r-1)p < i \leq rp.$$

Mọi lịch khả thi có thể đổi tên người chơi và nhóm để thỏa cách sắp này. Những Các mệnh đề đơn vị đó loại một phần lớn quỹ đạo mà hầu như không tăng chi phí giải [50].

7.4.2. Tiền tố sử dụng và thứ tự ưu tiên giá trị

Giả sử u_b cho biết khung chứa b được dùng. Vì tên các khung chứa có thể hoán vị, áp đặt

$$u_{b+1} \rightarrow u_b.$$

Các khung chứa được dùng tạo thành một tiền tố. Với các giá trị đối xứng, thứ tự ưu tiên giá trị yêu cầu lần xuất hiện đầu của giá trị nhỏ hơn đứng trước lần xuất hiện đầu của giá trị lớn hơn. Hai mẫu này cùng chọn thứ tự theo “lần sử dụng đầu”, nhưng tiền tố sử dụng chỉ cần một dãy Boolean nên thường rẻ hơn.

Mệnh đề 7.1 (Tiền tố sử dụng không làm mất lớp nghiệm). *Giả sử các khung chứa có cùng miền, cùng sức chứa và hàm mục tiêu không phụ thuộc vào tên khung. Với biến gán a_{ib} và biến sử dụng u_b , mọi nghiệm đều có một nghiệm cùng giá trị mục tiêu thỏa $u_{b+1} \rightarrow u_b$.*

Chứng minh. Cho $U = \{b : u_b = 1\}$ và $q = |U|$. Chọn một song ánh $\pi : U \rightarrow \{1, \dots, q\}$, rồi mở rộng π thành một hoán vị trên toàn bộ tập khung chứa. Định nghĩa

$$a'_{i,\pi(b)} = a_{ib}, \quad u'_{\pi(b)} = u_b.$$

Mọi biến thuộc bản thân vật i , như tọa độ, phép quay hay kích thước, được giữ nguyên. Vì sức chứa và các ràng buộc gán được phát biểu đồng nhất cho mọi khung, đổi tên theo π bảo toàn tính khả thi; vì hàm mục tiêu không đọc chỉ số khung, giá trị mục tiêu cũng không đổi. Trong nghiệm mới, $u'_1 = \dots = u'_q = 1$ và $u'_{q+1} = \dots = 0$, nên toàn bộ implication $u'_{b+1} \rightarrow u'_b$ đúng. $\square$

Chúng minh trên cũng chỉ rõ điều kiện không thỏa: nếu khung chứa có sức chứa, chi phí hoặc miền khác nhau thì hoán vị tùy ý không còn là đối xứng. Khi đó chỉ được áp dụng tiền tố trong từng lớp khung chứa thật sự đồng nhất.

7.4.3. Lex-leader

Với hai vector đại diện các đối tượng hoán vị được,

$$A = \langle a_1, \dots, a_m \rangle, \quad B = \langle b_1, \dots, b_m \rangle,$$

ràng buộc $A \leq_{\text{lex}} B$ chọn thứ tự từ điển. Một phép biểu diễn dùng các biến bằng nhau trên tiền tố e_i :

$$e_i \leftrightarrow e_{i-1} \wedge (a_i \leftrightarrow b_i), \quad e_0 = \top,$$

và cấm $e_{i-1} \wedge a_i \wedge \neg b_i$. *Lex-leader* (ràng buộc dẫn đầu theo thứ tự từ điển) mạnh hơn tiền tố sử dụng nhưng tạo thêm một chuỗi biến phụ cho mỗi cặp được so sánh.

7.4.4. Neo hình học và phản xạ

Với khung chứa hình chữ nhật, các phép phản xạ hoặc quay toàn bố cục có thể bảo toàn nghiệm. Một vật chuẩn—thường là vật lớn nhất theo diện tích rồi theo cạnh dài—được neo vào nửa trái hoặc một góc chuẩn:

$$2x_a + w_a \leq W.$$

Điều kiện phải khớp với nhóm đối xứng thật. Nếu khung chứa, miền quay hoặc hàm mục tiêu không bất biến qua phản xạ thì ràng buộc này không hợp lệ.

Mỗi ràng buộc phá vỡ đối xứng đối cấu trúc mệnh đề, thứ tự quyết định và các xung đột học được. Nên so sánh ít nhất ba cấu hình: không phá vỡ đối xứng, ràng buộc rẽ và ràng buộc mạnh. Kết luận dựa trên số bài toán giải được và thời gian, không chỉ trên số mô hình bị loại.

7.5. Biến quan hệ cho các phép tuyển

Nhiều bài toán có ràng buộc dạng

$$A \vee B \vee C \vee D.$$

Trong bài toán đóng gói, hai hình i, j không chồng lấn khi ít nhất một quan hệ đúng:

$$L_{ij} \vee L_{ji} \vee B_{ij} \vee B_{ji},$$

trong đó L_{ij} nghĩa là i nằm trái j , còn B_{ij} nghĩa là i nằm dưới j . Trong lập lịch, hai tác vụ dùng cùng tài nguyên cần

$$P_{ij} \vee P_{ji},$$

với P_{ij} nghĩa là i kết thúc trước khi j bắt đầu.

Witness variable (biến làm chứng) là biến chỉ ra nhánh nào của phép tuyển được chọn. Trong mô hình này, biến quan hệ đóng vai trò biến làm chứng cho một nhánh của phép tuyển. Chúng rút ràng buộc toàn cục thành các mệnh đề cục bộ quanh cặp đối tượng. Khi một quan hệ được chọn, lựa chọn này kích hoạt một quan hệ suy diễn kéo theo trên tọa độ hoặc thời gian; khi miền tọa độ loại quan hệ đó, mệnh đề tuyển buộc bộ giải chọn quan hệ còn lại.

7.5.1. Biến quan hệ và giá trị hỗ trợ

Từ góc nhìn chuyển CSP sang SAT, biến quan hệ nằm giữa Direct Encoding và mã hóa hỗ trợ. Mô hình không chọn toàn bộ giá trị của hai biến; phép biểu diễn chỉ chọn một *lớp* cặp giá trị, chẳng hạn “trái”, “dưới” hoặc “trước”. Mỗi lớp sau đó được liên kết sang các literal biên của miền. Đây là một phép phân tích thành nhân tử:

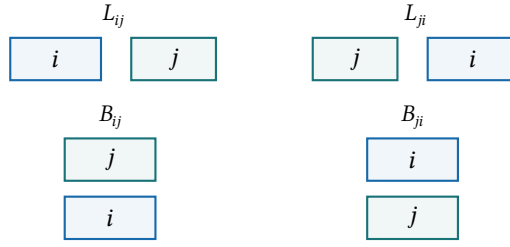
$$\text{bảng cặp giá trị hợp lệ} = \bigcup_{q \in Q} \text{lớp hỗ trợ}(q).$$

Nếu $|Q|$ nhỏ và mỗi lớp được mô tả bằng vài bất đẳng thức, các biến quan hệ nén một bảng cấm lớn. Nếu các lớp chồng lấn quá nhiều hoặc không có mô tả ngắn, biến quan hệ chỉ thêm một tầng trung gian.

7.6. Order Encoding và các literal so sánh

Cho biến nguyên $z \in \{0, \dots, U\}$. Order Encoding dùng

$$o_{z,t} = 1 \iff z \geq t, \quad t = 1, \dots, U.$$



Hình 7.3.: Bốn lớp hỗ trợ của quan hệ không chồng lấn giữa hai hình chữ nhật. Một literal quan hệ làm chứng cho cả một lớp cặp tọa độ.

Tính đơn điệu được biểu diễn bởi

$$o_{z,t+1} \rightarrow o_{z,t}.$$

Một cận $z \geq a$ là mệnh đề đơn vị $o_{z,a}$; cận $z \leq b$ là $\neg o_{z,b+1}$. Phép so sánh dịch chuyển

$$x_i + w_i \leq x_j$$

trở thành các mệnh đề cấm mọi cặp ngưỡng mâu thuẫn. Khi w_i là hằng, cấu trúc dịch chuyển này có tính địa phương cao và phù hợp với bài toán đóng gói (packing problem) [47, 49, 74].

Với lập lịch, cùng kỹ thuật biểu diễn

$$s_i + p_i \leq s_j.$$

Nhờ vậy, biến quan hệ P_{ij} có thể liên kết trực tiếp với các literal thứ tự của s_i, s_j , thay vì tạo một mạch cộng tổng quát [53].

Cho $z \in \{0, 1, 2, 3, 4\}$. Direct Encoding có $d_a \leftrightarrow (z = a)$; Order Encoding có $o_t \leftrightarrow (z \geq t)$. Hai góc nhìn được nối bởi

$$d_a \leftrightarrow o_a \wedge \neg o_{a+1}, \quad o_0 \equiv \top, \quad o_5 \equiv \perp.$$

Nếu bộ giải suy ra $o_3 = 1, o_4 = 0$, các mệnh đề liên kết buộc $d_3 = 1$, rồi ràng buộc đúng một trên $d_0, \dots, d_4$ loại mọi giá trị khác. Theo chiều ngược, $d_2 = 1$ buộc $o_1 = o_2 = 1$ và $o_3 = o_4 = 0$, khiến các ràng buộc cận đọc được ngay $z = 2$. Ví dụ này cho thấy lợi ích của liên kết hai chiều không nằm ở số mô hình mà ở đường truyền suy luận giữa hai giao diện.

7.7. Liên kết một chiều hay hai chiều

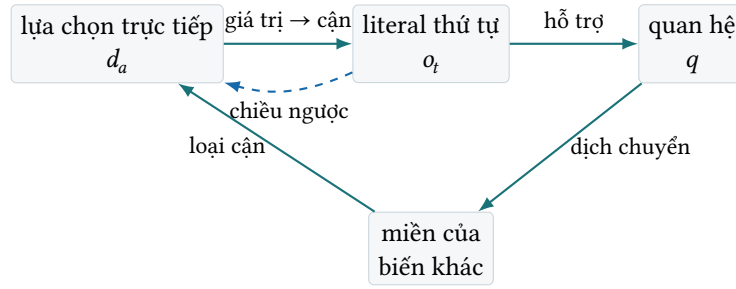
Giả sử q là biến quan hệ và Φ là công thức miền gốc biểu diễn quan hệ đó. Có ba mức liên kết:

$$q \rightarrow \Phi, \quad \Phi \rightarrow q, \quad q \leftrightarrow \Phi.$$

Chiều đầu đủ cho tính đúng nếu q chỉ được dùng làm biến làm chứng được chọn: bộ giải không thể đặt $q = 1$ khi quan hệ hình học sai. Chiều ngược giúp miền tọa độ suy ra q ; liên kết hai chiều cho phép lan truyền qua cả hai biểu diễn.

Lựa chọn phụ thuộc vào cách mô-đun sử dụng đọc biến:

- Nếu mệnh đề $q_1 \vee \dots \vee q_r$ yêu cầu một biến làm chứng và không nơi nào đọc $\neg q$, quan hệ kéo theo tiến thường đủ.
- Nếu $\neg q$ được dùng để kết luận quan hệ đối nghịch, phải có đặc tả mạnh hơn.



Hình 7.4.: Vòng liên kết trong phép biểu diễn kết hợp. Thiếu một chiều có thể làm đứt một đường lan truyền đơn vị dù công thức vẫn đúng theo phép chiếu.

- Nếu nhiều ràng buộc và hàm mục tiêu chia sẻ q , quan hệ tương đương giúp tránh các trạng thái phụ không mang nghĩa.

Ví dụ 7.2. Với hai tác vụ cùng máy, đặt $P_{ij} \vee P_{ji}$. Hai mệnh đề liên kết

$$P_{ij} \rightarrow (s_i + p_i \leq s_j), \quad P_{ji} \rightarrow (s_j + p_j \leq s_i)$$

đã bảo đảm chúng không chồng lấn. Nếu quan hệ trước-sau từ một ràng buộc khác cho biết $s_i + p_i \leq s_j$, chiều ngược $(s_i + p_i \leq s_j) \rightarrow P_{ij}$ có thể truyền quan hệ này sang các mô-đun khác.

7.7.1. Lan truyền qua vòng liên kết

Phép biểu diễn kết hợp tạo một vòng suy luận:

lựa chọn chính xác $\rightarrow$ các ngưỡng $\rightarrow$ quan hệ $\rightarrow$ ngưỡng của biến khác.

Mỗi chiều liên kết đóng một đường suy luận trong vòng trên. Phép đánh giá bắt đầu bằng danh sách các suy luận miền mong muốn, sau đó tìm tập chiều tối thiểu tạo được chúng. Đặc tả này có thể kiểm bằng các phép gán một phần và lan truyền đơn vị như trong Chương 2.

Việc giữ nhiều góc nhìn cũng liên quan đến biểu diễn lại có cấu trúc: biến phụ có giá trị khi đặt tên cho một lớp hỗ trợ được nhiều ràng buộc dùng lại, thay vì chỉ thay một biểu thức xuất hiện một lần [40].

7.8. Phép quay và kích thước có điều kiện

Khi vật i được phép quay, biến ρ_i chọn hướng đặt:

$$(w_i(\rho_i), h_i(\rho_i)) = \begin{cases} (w_i, h_i), & \rho_i = 0, \\ (h_i, w_i), & \rho_i = 1. \end{cases}$$

Các literal quan hệ được chia sẻ cho hai hướng đặt và được liên kết với phép so sánh dịch chuyển dưới điều kiện gác ρ_i, ρ_j . Ví dụ, L_{ij} sinh bốn họ mệnh đề, mỗi họ ứng với một tổ hợp hướng đặt còn hợp lệ. Các tổ hợp không vừa khung chứa bị loại bằng mệnh đề đơn vị hoặc mệnh đề nhị phân trước khi sinh quan hệ cặp.

Cách tổ chức này tách ba tầng:

hướng đặt $\rightarrow$ kích thước hiệu dụng $\rightarrow$ quan hệ không chồng lấn.

Cấu trúc này cũng cho phép tái sử dụng cùng biến ρ_i trong cận miền, quy tắc trội và bộ giải mã.

7.9. Đồ thị xung đột và tính địa phương

Đồ thị xung đột $H = (V, E)$ chọn chính xác các cặp cần biến quan hệ: $\{i, j\} \in E$ khi hai đối tượng có thể cạnh tranh cùng tài nguyên, chồng lấn hình học hoặc vi phạm một cận cục bộ.

Trong lập lịch, cạnh xuất hiện khi hai tác vụ có thể cùng dùng một tài nguyên và miền thời gian của chúng giao nhau. Trong bài toán đóng gói, mọi cặp vật cùng khung chứa là ứng viên, nhưng cận miền và quan hệ trội có thể cố định trước một số quan hệ. Trong phân bố tần số, cạnh là cặp máy phát có yêu cầu khoảng cách nhân.

Thiết kế theo đồ thị đem lại hai lợi ích: kích thước tỉ lệ với số tương tác thật, và mệnh đề học được thường nằm quanh một lân cận nhỏ. Đây là một tiêu chí cụ thể về mức độ phù hợp với bộ giải, thay vì chỉ nói phép biểu diễn “gọn”.

7.10. Kiến trúc kết hợp

Một mô hình SAT lớn nên ghép các mô-đun theo thứ tự sau:

1. chọn biến chính loại các chỉ số không mang nghĩa;
2. tạo miền thứ tự/trực tiếp cho thời gian, tọa độ hoặc nhân;
3. tạo biến quan hệ trên đồ thị xung đột;
4. liên kết mỗi quan hệ với các literal miền;
5. thêm ràng buộc đếm hoặc Shared Counter cho tài nguyên;
6. thêm các ràng buộc phá vỡ đối xứng có chứng minh quỹ đạo;
7. gán hàm mục tiêu bằng các giả định hoặc mệnh đề mềm.

Các nghiên cứu về lập lịch, đóng gói dải/khung chứa và Social Golfer trong Tuyên [86] cho thấy ba lựa chọn đầu—biến chính, quan hệ và đối xứng—quyết định cấu trúc CNF trước khi chọn bộ giải. Các chương sau dùng chính kiến trúc này cho bốn họ bài toán.

7.11. Bài học thiết kế

1. Loại đối xứng biểu diễn trước khi thêm các mệnh đề phá vỡ đối xứng.
2. Một breaker hợp lệ phải giữ ít nhất một đại diện của mọi quỹ đạo.
3. Biến quan hệ là biến làm chúng có ngữ nghĩa, giúp cục bộ hóa phép tuyến.
4. Liên kết mạnh đến đâu phụ thuộc vào cách biến quan hệ được đọc.
5. Order Encoding làm các cận dịch chuyển trở thành mệnh đề đơn giản.
6. Đồ thị xung đột xác định nơi thật sự cần biến phụ.

Câu hỏi nghiên cứu

1. Chứng minh tiền tố sử dụng giữ một đại diện khi các khung chứa hoàn toàn giống nhau.
2. Viết liên kết một chiều cho bốn quan hệ không chồng lấn của hai hình.

3. Cho một ràng buộc phản xạ trong bài toán đóng gói (packing problem) dài và nêu điều kiện để ràng buộc đó hợp lệ.

PHẦN III

Các họ bài toán tối ưu tổ hợp

8 Lập lịch và cân bằng dây chuyền

TÓM TẮT CHƯƠNG

Một phép biểu diễn cho bài toán thời gian chỉ thực sự rõ khi trả lời trực tiếp ba câu hỏi: quyết định nằm ở mốc nào, điều gì phải liên tục giữa các mốc, và cận mục tiêu được thay đổi ra sao trong quá trình giải.

8.1. Từ SAT theo mốc thời gian đến lazy clause generation

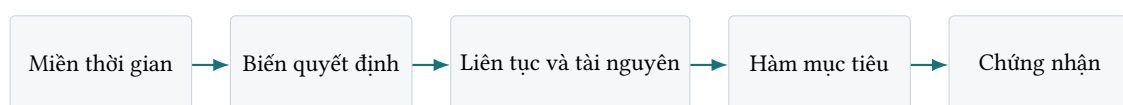
Lập lịch là một trong những miền sớm cho thấy SAT có thể giải các bài toán ngoài logic thuần túy. Các thí nghiệm SAT về lập lịch từ thập niên 1990 đã dùng quyết định theo mốc thời gian và mệnh đề xung đột để so sánh với phương pháp tìm kiếm chuyên dụng [26]. Từ đó, văn liệu phát triển theo ba hướng:

1. biên dịch toàn bộ ràng buộc gán, thời gian và tài nguyên sang SAT;
2. dùng MaxSAT/PB để biểu diễn mức phạt, ưu tiên hoặc số ca vi phạm;
3. tích hợp lan truyền miền với học mệnh đề, điển hình là *lazy clause generation* (sinh mệnh đề lười, LCG).

LCG giữ biến miền và bộ lan truyền của CP, đồng thời gán mỗi phép loại giá trị với một giải thích dạng mệnh đề; phân tích xung đột sau đó học trên các giải thích này. Trên RCPSP/max, kiến trúc đó kết hợp lan truyền lập lịch toàn cục với học mệnh đề [70]. Ở một hướng khác, MaxSAT bộ phận có trọng số cho phép tách độ phủ/tính hợp lệ thành các mệnh đề cứng và ưu tiên trong lập lịch nhân viên thành các mệnh đề mềm [30].

Bảng 8.1.: Các mô hình giải chính xác cho bài toán thời gian.

Mô hình	Trạng thái thời gian	Cơ chế tài nguyên	Cận chủ đạo
SAT theo mốc thời gian	Mốc rời rạc	Xung đột/ràng buộc đếm	UNSAT theo khung thời gian
SAT thú tự	Ngưỡng thời gian	Quan hệ + cận dịch chuyển	Học mệnh đề
MaxSAT/PB	Như SAT + literal chi phí	Cứng/mềm hoặc PB	Lỗi/cận chi phí
SMT	Biến nguyên/thực	Phép tuyến + lý thuyết	Lan truyền lý thuyết
CP/LCG	Miền và khoảng	Bộ lan truyền toàn cục	Lọc miền + mệnh đề
MIP	Biến nhị phân/liên tục	Bất đẳng thức tuyến tính	Nói lỏng quy hoạch tuyến tính



Hình 8.1.: Lộ trình lập luận của chương: từ Miền thời gian, qua Biến quyết định, Liên tục và tài nguyên và Hàm mục tiêu, đến Chứng nhận.

Vì vậy, một phép biểu diễn lập lịch được đánh giá trên cả kích thước, sức lan truyền và biểu diễn đối chúng tương ứng: SAT theo mốc thời gian so với MIP mở rộng theo thời gian; SAT thứ tự so với logic sai phân/SMT; bộ đếm chuỗi dùng chung so với bộ lan truyền SEQUENCE; và hàm mục tiêu MaxSAT so với PB hay hàm mục tiêu tuyến tính.

8.2. Kiến trúc chung của mô hình thời gian

Các bài toán trong chương có bề mặt khác nhau: xếp tác vụ không ngắt trên các máy đồng nhất, phân ca điều dưỡng, lập lịch cuộc họp B2B với cân bằng thời gian chờ, cân bằng dây chuyền theo đỉnh công suất và điều chỉnh lịch tàu. Tuy nhiên, phép biểu diễn của chúng đều có thể tách thành bốn lớp:

1. **gán**: tác vụ, người hoặc chuyển tàu sử dụng tài nguyên nào;
2. **thời gian/thứ tự**: thời điểm bắt đầu, kết thúc hoặc thứ tự trước-sau;
3. **liên tục/chuỗi**: các quyết định cục bộ ghép thành một khoảng hoặc một chuỗi hợp lệ;
4. **hàm mục tiêu**: thời điểm hoàn tất toàn lịch, đỉnh công suất, độ trễ hay số vi phạm.

Phân tách kiến trúc theo lớp không chỉ giúp trình bày mạch lạc, mà còn cho phép thay thế một mô-đun mà không đổi ngữ nghĩa của các mô-đun còn lại. Chẳng hạn, có thể giữ nguyên biến gán máy và thay phép biểu diễn cho tính không ngắt; hoặc giữ nguyên mô hình khả thi và thay vòng lặp tối ưu bằng SAT tăng dần hay MAXSAT.

Mỗi biến phụ phải thuộc về một lớp rõ ràng. Nếu một biến vừa được dùng để suy thời điểm, vừa được diễn giải như trạng thái hoạt động, cần có liên kết hai chiều hoặc một phép giải mã chúng minh được.

8.3. Lập lịch không ngắt trên các tài nguyên đồng nhất

8.3.1. Mô hình

Cho tập tác vụ $J = \{1, \dots, n\}$ và m tài nguyên đồng nhất. Tác vụ i có thời điểm sẵn sàng r_i , thời lượng e_i và hạn hoàn tất d_i . Thời điểm bắt đầu thuộc miền

$$T_i = \{r_i, r_i + 1, \dots, d_i - e_i\}.$$

Nếu $s_i \in T_i$, tác vụ chiếm tài nguyên trong nửa khoảng $[s_i, s_i + e_i)$. Hai tác vụ được gán cùng tài nguyên không được có hai khoảng giao nhau.

Một mô hình gọn dùng hai họ biến:

$$y_{ij} = 1 \iff i \text{ được gán cho tài nguyên } j, \quad z_{it} = 1 \iff i \text{ hoạt động tại } t.$$

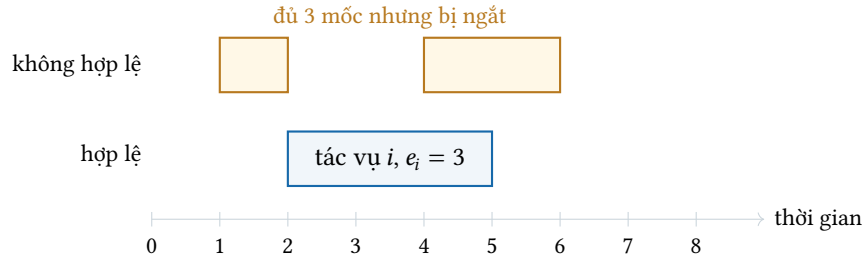
Với mỗi tác vụ, các biến y_{ij} thỏa

$$\text{ExactlyOne}(y_{i1}, \dots, y_{im}).$$

Xung đột tài nguyên tại mốc t được loại bởi

$$\neg z_{it} \vee \neg y_{ij} \vee \neg z_{i't} \vee \neg y_{i'j}, \quad i < i'.$$

Mệnh đề này thể hiện xung đột có điều kiện: mệnh đề chỉ kích hoạt khi hai tác vụ cùng chọn tài nguyên j .



Hình 8.2.: Ràng buộc đếm $\sum_t z_{it} = 3$ không phân biệt một khối liên tục với ba mốc rời; tính không ngắt phải biểu diễn vị trí bắt đầu và hai biên khối.

8.3.2. Từ ràng buộc đếm đến một khối liên tục

Điều kiện

$$\sum_t z_{it} = e_i$$

chỉ bảo đảm đủ số mốc hoạt động; cơ chế này không ngăn các mốc bị phân tách thành nhiều đoạn. *Non-preemption* (tính không ngắt: tác vụ phải chạy liên tục từ lúc bắt đầu đến lúc kết thúc) đòi hỏi mạnh hơn:

$$z_{it} = 1 \iff s_i \leq t < s_i + e_i.$$

Direct Encoding có thể tạo một biến b_{is} cho từng thời điểm bắt đầu $s \in T_i$. Trước hết phải đóng lựa chọn thời điểm bắt đầu bằng

$$\text{ExactlyOne}(b_{is} : s \in T_i).$$

Trong CNF, phần ít nhất một là mệnh đề $\bigvee_{s \in T_i} b_{is}$; phần nhiều nhất một được sinh bằng biểu diễn từng cặp, Sequential Counter hoặc một mô-đun AMO khác tùy kích thước T_i . Sau đó dùng

$$b_{is} \rightarrow z_{it} \quad \text{với } s \leq t < s + e_i, \quad b_{is} \rightarrow \neg z_{it} \quad \text{ở ngoài khoảng.}$$

Cách này dễ hiểu nhưng lặp lại nhiều literal cho các thời điểm bắt đầu lân cận. Phép biểu diễn gọn thay phần lặp bằng các biến tiền tố/hậu tố có ngữ nghĩa “tất cả bằng 0” hoặc “tất cả bằng 1”. Hai thời điểm bắt đầu kế tiếp chia sẻ gần như toàn bộ cấu trúc; khác biệt của chúng chỉ nằm ở hai biên của khối.

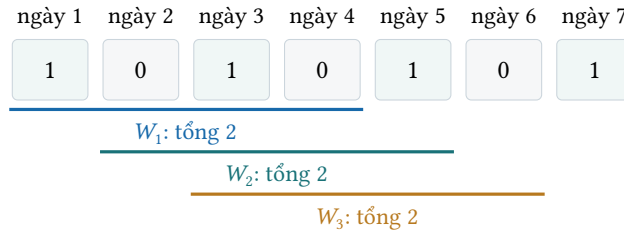
Mệnh đề 8.1 (Đúng đắn của phép biểu diễn theo khối). *Giả sử mô-đun khối bảo đảm: (i) tồn tại đúng một vị trí bắt đầu; (ii) từ vị trí đó có đúng e_i giá trị liên tiếp bằng 1; và (iii) mọi vị trí trước và sau khối đều bằng 0. Khi đó phép gán cho các biến z_{it} tương ứng duy nhất với một khoảng xử lý không ngắt hợp lệ của tác vụ i .*

Chứng minh. Từ (i), chọn vị trí bắt đầu s_i . Điều kiện (ii) cho $z_{it} = 1$ trên $[s_i, s_i + e_i)$; điều kiện (iii) loại mọi giá trị 1 khác. Chiều ngược lại, từ một khoảng hợp lệ, đặt biến bắt đầu ở đầu khoảng và gán các biến tiền tố/hậu tố theo định nghĩa của chúng. Mọi mệnh đề của mô-đun đều được thỏa. $\square$

Cho $J_1 = (r = 0, e = 3, d = 6)$, $J_2 = (0, 2, 5)$, $J_3 = (1, 2, 6)$. Một lịch hợp lệ là

$$M_1 : J_1[0, 3) \quad J_3[3, 5), \quad M_2 : J_2[0, 2).$$

Với $t = 1$, $z_{1,1} = z_{2,1} = 1$ nhưng hai tác vụ dùng máy khác nhau nên mệnh đề xung đột có điều kiện vẫn thỏa. Nếu đổi $y_{2,2}$ thành $y_{2,1}$, mệnh đề tại $t = 1, j = 1$ trở thành $\neg z_{1,1} \vee \neg y_{1,1} \vee \neg z_{2,1} \vee \neg y_{2,1}$ và bị vi phạm. Ví dụ tách rõ hai lớp: z nói khi nào tác vụ chạy, y nói tác vụ chạy ở vị trí nào.



Hình 8.3.: Ba cửa sổ ALSC ($w = 4, k = 2$) trên bảy ngày. Mỗi cửa sổ kế nhau chỉ thay một ngày, nên các ngưỡng ở phần giao có thể dùng lại.

Kết quả của Kieu và To [53] cho thấy lợi ích chính của phép mã hóa gọn nằm ở việc tránh lặp cấu trúc liên tục. Cấu hình kết hợp biểu diễn theo khối với phá vỡ đối xứng và một phép biểu diễn giả Boolean thích hợp đạt tổng thời gian thấp nhất trong nhiều nhóm bài toán của nghiên cứu đó. Điều cần rút ra cho thiết kế là: *tính không ngắt nên là một mô-đun độc lập*, thay vì được ngấm suy từ một ràng buộc đếm.

8.4. Ràng buộc chuỗi trong lập lịch điều dưỡng

8.4.1. Cửa sổ trượt

Với dãy Boolean $\Omega = (x_1, \dots, x_n)$, một ràng buộc chuỗi cận dưới yêu cầu mọi cửa sổ độ rộng w chứa ít nhất k giá trị đúng:

$$\text{ALSC}(\Omega, w, k) = \bigwedge_{i=0}^{n-w} \left(\sum_{j=i+1}^{i+w} x_j \geq k \right).$$

Nếu biểu diễn từng cửa sổ bằng một phép biểu diễn ràng buộc đếm độc lập, hai cửa sổ kế nhau chia sẻ $w - 1$ biến gốc nhưng không chia sẻ biến phụ. Cấu trúc chuỗi vì thế đặt ra bài toán *tái sử dụng biểu thức con*.

Chia một cửa sổ thành hai phần A và B . Ta có

$$\sum_{x \in A \cup B} x \geq k \iff \bigwedge_{q=1}^k \left[\left(\sum_{x \in A} x \geq q \right) \vee \left(\sum_{x \in B} x \geq k - q + 1 \right) \right].$$

Mỗi mệnh đề ngưỡng trong ngoặc được biểu diễn bởi một thanh ghi chia sẻ. Khi cửa sổ trượt sang phải, phần lớn thanh ghi vẫn còn nguyên.

Ví dụ 8.2. Xét $x_3 + x_4 + x_5 + x_6 \geq 2$, chia thành hai khối $A = (x_3, x_4)$ và $B = (x_5, x_6)$. Nếu $R_{A,q}$ mang nghĩa “ A có ít nhất q giá trị đúng”, bộ nối chỉ cần

$$(R_{A,1} \vee R_{B,2}) \wedge (R_{A,2} \vee R_{B,1}).$$

Các thanh ghi của B tiếp tục được dùng trong cửa sổ kế tiếp.

8.4.2. Đưa vào mô hình phân ca

Đặt $x_{ids} = 1$ khi điều dưỡng i làm loại ca s ở ngày d . Mỗi cặp (i, d) nhận đúng một trạng thái. Những quy tắc như “trong mọi w ngày có ít nhất k ngày nghỉ” trở thành ALSC; quy tắc at-most được đối ngẫu:

$$\sum_{j=1}^w x_j \leq q \iff \sum_{j=1}^w (1 - x_j) \geq w - q.$$

Nhờ đó một bản cài đặt có thể dùng cùng một mô-đun cận dưới cho cả hai hướng.

Trong một mô hình phân ca đầy đủ, ràng buộc chuỗi chỉ là một lớp. Lớp độ phủ bảo đảm số nhân viên mỗi ca; ràng buộc hợp đồng giới hạn tổng giờ và số cuối tuần; ưu tiên tạo các chi phí mềm. Phép biểu diễn của sổ chia sẽ tác động trực tiếp lên lớp chuỗi, còn MaxSAT có trọng số là một lựa chọn cho hàm mục tiêu của toàn mô hình [30]. Ma trận thiết kế tách các lớp để đo riêng ảnh hưởng của ALSC, độ phủ và ưu tiên.

Trên 28 bài toán *Nurse Rostering* (lập lịch điều dưỡng), với quy mô đến 120 điều dưỡng và 24 tuần, ALSCE giải toàn bộ tập thử trong nghiên cứu của Truong và cộng sự [82]; bài toán lớn nhất được báo cáo ở mức 26,01 giây. Kết quả này minh họa hiệu quả của kỹ thuật chia sẽ thanh ghi giữa các cửa sổ chồng lấn.

8.5. Lập lịch cuộc họp B2B với cân bằng thời gian chờ

8.5.1. Bài toán B2B và mục tiêu khoảng thời gian chờ

Bài toán lập lịch cuộc họp doanh nghiệp (Business-to-Business Meeting Scheduling Problem, B2B) xếp lịch cho tập các cuộc họp đối tác M vào tập mốc thời gian rời rạc $T = \{1, \dots, h\}$ và K phòng họp/địa điểm dùng chung [15, 66]. Mỗi cuộc họp $m \in M$ được thực hiện bởi đúng hai đại biểu, ký hiệu $\pi(m) \subseteq P$ với $|\pi(m)| = 2$, và có miền mốc khởi tạo $D_m^0 \subseteq T$ tương ứng với phiên làm việc, mốc cố định và mốc cấm của hai đại biểu. Đặt $E \subseteq M \times M$ là tập các cung thứ tự tiên quyết giữa các cuộc họp. Một lịch xếp $S : M \rightarrow T$ là hợp lệ nếu $S(m) \in D_m^0$ và thỏa mãn các điều kiện:

$$S(m) \neq S(m') \quad \forall m, m' \in M, m \neq m' : \pi(m) \cap \pi(m') \neq \emptyset, \quad (8.1)$$

$$|\{m \in M \mid S(m) = t\}| \leq K \quad \forall t \in T, \quad (8.2)$$

$$S(i) < S(j) \quad \forall (i, j) \in E. \quad (8.3)$$

Với mỗi đại biểu $p \in P$, đặt $M_p = \{m \in M \mid p \in \pi(m)\}$ là tập cuộc họp của đại biểu đó. Đặt $P^* = \{p \in P \mid |M_p| \geq 2\}$ là tập các đại biểu có ít nhất hai cuộc họp. Đối với $p \in P^*$, mốc bận đầu tiên $F_p(S)$ và mốc bận cuối cùng $L_p(S)$ trong lịch S được xác định bởi:

$$F_p(S) = \min_{m \in M_p} S(m), \quad L_p(S) = \max_{m \in M_p} S(m).$$

Do các cuộc họp của đại biểu p không chồng lấn, số mốc thời gian trống nằm hoàn toàn giữa mốc đầu và mốc cuối—gọi là *thời gian chờ nội bộ* (internal idle slots)—được tính bằng:

$$I_p(S) = \begin{cases} L_p(S) - F_p(S) + 1 - |M_p|, & p \in P^*, \\ 0, & |M_p| \leq 1. \end{cases} \quad (8.4)$$

Khác với mô hình truyền thống đếm tổng số nhóm nghỉ (*break groups*) [15], chỉ số $I_p(S)$ phân biệt rõ khoảng chờ dài (như chờ 5 mốc) với khoảng chờ ngắn (chờ 1 mốc). Hàm mục tiêu tối ưu hóa khoảng thời gian chờ nội bộ giữa các đại biểu:

$$\Delta_I(S) = \max_{p \in P^*} I_p(S) - \min_{p \in P^*} I_p(S). \quad (8.5)$$

Mục tiêu này hướng tới sự cân bằng công bằng về thời gian chờ giữa các đại biểu, lấy cảm hứng từ các chuẩn cân bằng tải trong bài toán BACP [22]. Tổng thời gian chờ $I_\Sigma(S) = \sum_{p \in P} I_p(S)$ được báo cáo như một chỉ số hỗ trợ.

8.5.2. Tiền xử lý bảo toàn nghiệm và lọc miền

Trước khi sinh công thức CNF, thuật toán lọc miền [64] thu hẹp miền mốc của từng cuộc họp bằng cách áp dụng lập cho đến điểm bất động ba quy tắc lọc:

1. **Cận thứ tự tiên quyết:** Với mỗi quan hệ (i, j, d) trong tập quan hệ $\mathcal{R}_f \in \{\mathcal{R}_E, \mathcal{R}_{E^*}\}$ (trong đó $\mathcal{R}_E = \{(i, j, 1) \mid (i, j) \in E\}$ và $\mathcal{R}_{E^*}$ là bao đóng bắc cầu gán khoảng cách đường đi dài nhất ℓ_{ij}), loại $t \in D_i$ nếu $t + d > \max D_j$, và loại $u \in D_j$ nếu $\min D_i + d > u$.
2. **Nhất quán ghép cặp đại biểu:** Với mỗi đại biểu p , dựng đồ thị hai phía giữa M_p và miền mốc khả thi. Sử dụng thuật toán lọc ALLDIFFERENT dựa trên ghép cặp cực đại [69] để xóa phân gán cuộc họp–mốc nào không thuộc bất kỳ ghép cặp hoàn hảo nào của M_p .
3. **Mốc bão hòa:** Nếu mốc t đã bị ép chọn bởi đúng K cuộc họp cố định, loại mốc t khỏi miền của tất cả cuộc họp còn lại.

Mệnh đề 8.3 (Tính bảo toàn nghiệm của lọc miền B2B). *Thuật toán lọc miền dùng sau số bước hữu hạn, loại bỏ duy nhất các giá trị không thể thuộc bất kỳ lịch hợp lệ nào, bảo toàn trọn vẹn tập nghiệm khả thi, và chỉ thông báo vô nghiệm khi bài toán thực sự không có nghiệm.*

Chứng minh. Mỗi bước loại giá trị đều dựa trên các điều kiện cần về tính khả thi: vi phạm cận thứ tự tiên quyết, không có hỗ trợ trong ghép cặp đại biểu, hoặc vượt quá sức chứa địa điểm K . Vì miền ban đầu là hữu hạn, quá trình xóa giá trị phải dừng sau tối đa $\sum_m |D_m^0|$ bước. $\square$

Phép biểu diễn REDUCED áp dụng biến gán $x_{m,t}$ chỉ trên miền mốc đã lọc $A_m = D_m$, thay vì miền đầy đủ $A_m = T$ của mô hình gốc ORG-MAXSAT [15].

8.5.3. Biểu diễn thứ tự tiên quyết bằng hậu tố thừa chia sẻ (SPARSESUFFIX)

Xét quan hệ tiên quyết $(i, j, d) \in \mathcal{R}_g$. Biểu diễn từng cặp (PAIRWISE) thêm các mệnh đề nhị phân $\neg x_{i,t} \vee \neg x_{j,u}$ với mọi $t \in A_i, u \in A_j$ thỏa $t + d > u$, có thể phát sinh $O(|A_i||A_j|)$ mệnh đề cho một quan hệ.

Phép biểu diễn hậu tố thừa (SPARSESUFFIX) [64] chỉ khởi tạo biến phụ tại các điểm cắt nội bộ thực sự xuất hiện trong miền tiên nhiệm. Với miền đã sắp thứ tự $A_i = (a_{i,0}, \dots, a_{i,n_i-1})$, giả sử các điểm cắt xuất hiện là $c_1 < c_2 < \dots < c_s$. Đặt $c_{s+1} = n_i$ và $q_{i,n_i} = \perp$, biến đại diện hậu tố được định nghĩa quy nạp:

$$c(i, j, u) = \min \{r \in \{0, \dots, n_i - 1\} \mid a_{i,r} > u - d\},$$

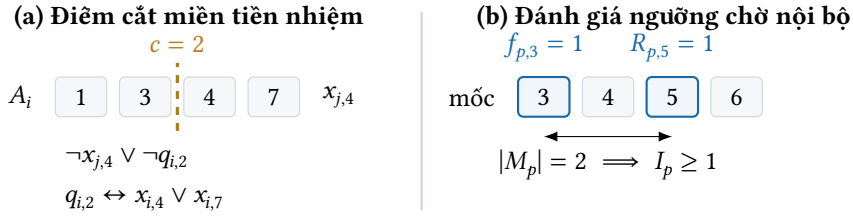
$$q_{i,c_t} \leftrightarrow \left(\bigvee_{r=c_t}^{c_{t+1}-1} x_{i,a_{i,r}} \right) \vee q_{i,c_{t+1}}, \quad (8.6)$$

$$x_{j,u} \rightarrow \neg q_{i,c(i,j,u)}. \quad (8.7)$$

Khi hai quan hệ tiên quyết cùng dẫn đến một điểm cắt c trên miền A_i , biến hậu tố $q_{i,c}$ được tái sử dụng hoàn toàn.

Mệnh đề 8.4 (Tương đương của SparseSuffix). *Phép biểu diễn SPARSESUFFIX và phép biểu diễn từng cặp PAIRWISE mô tả chính xác cùng một tập phân gán cuộc họp–mốc thời gian.*

Chứng minh. Từ Công thức (8.6), biến phụ $q_{i,c}$ đúng khi và chỉ khi cuộc họp i được gán cho mốc có chỉ số ít nhất là c . Mệnh đề kéo theo (8.7) do đó loại bỏ chính xác các cặp mốc (t, u) vi phạm điều kiện $t + d \leq u$. $\square$



Hình 8.4.: Kỹ thuật biểu diễn trong B2B: (a) Biến hậu tố thừa SPARSESUFFIX tại điểm cắt $c = 2$; (b) Xác định ngưỡng thời gian chờ nội bộ từ mốc bắt đầu và mốc kết thúc.

8.5.4. Mã hóa ngưỡng thời gian chờ từ nhị hoạt động

Để xác định $I_p(S)$, gọi $y_{p,t} \leftrightarrow \bigvee_{m \in M_p: t \in A_m} x_{m,t}$ là biến chỉ báo đại biểu p bận tại mốc t . Định nghĩa biến tích lũy tiền tố $P_{p,t} \leftrightarrow P_{p,t-1} \vee y_{p,t}$ và hậu tố $R_{p,t} \leftrightarrow y_{p,t} \vee R_{p,t+1}$. Mốc bắt đầu đầu tiên của đại biểu p được nhận diện bằng:

$$f_{p,t} \leftrightarrow y_{p,t} \wedge \neg P_{p,t-1}.$$

Gọi $\theta_{p,k}$ là biến mệnh đề đại diện cho điều kiện $I_p(S) \geq k$. Mối liên hệ giữa mốc đầu và mốc cuối được biểu diễn chính xác bằng:

$$f_{p,t} \rightarrow \left(\theta_{p,k} \leftrightarrow R_{p,t+|M_p|+k-1} \right) \quad \forall t \in T, 1 \leq k \leq U_p, \quad (8.8)$$

trong đó U_p là cận trên hợp lệ của thời gian chờ đối với đại biểu p .

Cuối cùng, với $U = \max_{p \in P^*} U_p$ và $1 \leq k \leq U$, biểu diễn định nghĩa biến ngưỡng cực đại $\Theta_k^{\max} \leftrightarrow \bigvee_{p \in P^*} \theta_{p,k}$, cực tiểu $\Theta_k^{\min} \leftrightarrow \bigwedge_{p \in P^*} \theta_{p,k}$, và biến mức chênh lệch:

$$g_k \leftrightarrow \Theta_k^{\max} \wedge \neg \Theta_k^{\min}, \quad \text{với } \sum_{k=1}^U g_k = \Delta_I(S). \quad (8.9)$$

8.5.5. Phương pháp giải và kết quả thực nghiệm

Phép biểu diễn B2B thu gọn được đánh giá thực nghiệm rộng rãi trên 126 bộ dữ liệu chuẩn chính thức (*All-126*) và 100 bộ dữ liệu biến đổi mật độ thứ tự (*Stress-100*) [64]. Thử nghiệm so sánh ba phương pháp tối ưu dựa trên công thức SAT/MaxSAT (UWrMaxSAT, NON-INCREMENTAL SAT, và INCREMENTAL SAT sử dụng Totalizer kết hợp giả định assumption) với các công cụ thương mại hàng đầu đại diện cho MIP (Gurobi, CPLEX) và CP (CP Optimizer).

Trên tập dữ liệu chuẩn *All-126*:

- Phép biểu diễn thu gọn giảm trung vị **40.3%** số mệnh đề, **55.9%** dung lượng bộ nhớ đỉnh và **14.0%** tổng thời gian chạy so với mô hình ORG-MaxSAT công bố [15].
- Riêng bước lọc miền (REDUCED) đóng góp giảm **24.1%** số biến gán và **16.2%** số mệnh đề.
- Biểu diễn hậu tố SPARSESUFFIX kết hợp với bao đóng bắc cầu CLOSURE- E^* giúp giảm số mệnh đề đến **5.5%** ở các mức mật độ thứ tự cao.
- Cả ba phương pháp SAT và MaxSAT đều giải thành công 100% các bài toán chính thức với thời gian trung vị vượt trội Gurobi MIP (0.521 giây); trong đó INCREMENTAL SAT đạt thời gian trung vị nhanh nhất (**0.227** giây).

8.6. Peak power: công suất đỉnh trong cân bằng dây chuyền

8.6.1. Mô hình đỉnh công suất

Trong SALBP-3PM, tác vụ i có thời lượng t_i , công suất w_i , được gán vào một trạm và bắt đầu trong chu kỳ dài c . Nếu $A_{i\tau}$ biểu diễn “tác vụ i đang hoạt động tại τ ”, tổng công suất là

$$W(\tau) = \sum_i w_i A_{i\tau}, \quad P = \max_{\tau=0}^{c-1} W(\tau).$$

Mục tiêu là tối thiểu hóa P .

Các biến trực tiếp cho quyết định gán và bắt đầu thuận tiện cho phép giải mã:

$$X_{ik} = 1 \iff i \text{ ở trạm } k, \quad S_{it} = 1 \iff i \text{ bắt đầu tại } t.$$

Các biến thứ tự cung cấp lớp lan truyền:

$$R_{ik} = 1 \iff \text{trạm}(i) \leq k, \quad T_{it} = 1 \iff \text{bắt đầu}(i) \leq t.$$

Giá trị chính xác xuất hiện tại điểm chuyển của chuỗi thứ tự. Nhờ vậy, ràng buộc AMO về phép gán giảm từ dạng từng cặp bậc hai xuống một chuỗi tuyến tính; quan hệ trước-sau giữa trạm hoặc giữa mốc thời gian cũng được viết bằng quan hệ kéo theo giữa các ngưỡng.

8.6.2. Bốn giao diện tối ưu

Cùng một phép biểu diễn tính khả thi có thể kết nối với hàm mục tiêu theo bốn cách:

Bảng 8.2.: Bốn giao diện tối ưu cho công suất đỉnh.

Cơ chế	Cách thay cận	Thành phần cần giữ cố định
Chặn bằng mệnh đề	Cấm mọi đỉnh không tốt hơn nghiệm đương nhiệm	Ngữ nghĩa của biến đỉnh
Tinh chỉnh PB	Siết $\sum_i w_i A_{i\tau} \leq B$ tại mọi τ	Phép biểu diễn giả Boolean
MaxSAT có trọng số	Đưa mức đỉnh vào mệnh đề mềm	Ảnh xạ chi phí với giá trị P
Biến chỉ báo thứ tự	Kích hoạt cận bằng các giả định	Mệnh đề cơ sở và chuỗi thứ tự

Biến chỉ báo thứ tự phù hợp với giải tăng dần: bộ giải được dựng một lần ở cận trên, sau mỗi nghiệm chỉ thay các giả định hoặc thêm mệnh đề chặn. Trong nghiên cứu của Kieu và cộng sự [51], phép biểu diễn tuần tự gọn kết hợp giải tăng dần chúng nhận được nhiều bài toán hơn cấu hình đối chiếu trên tập 89 bài kiểm chuẩn; lợi ích rõ nhất ở nhóm khó.

Mô-đun khả thi mô tả lịch hợp lệ qua một giao diện độc lập với tìm kiếm tuyến tính, tìm kiếm nhị phân hay MAXSAT. Lớp hàm mục tiêu đọc các biến đã có và cung cấp giao diện cận B . Cách tách này làm cho chứng minh tính đúng đắn ngắn hơn và thí nghiệm phân tích thành phần (ablation study) rõ hơn.

8.7. Điều chỉnh lịch tàu bằng MaxSAT-DDD

8.7.1. Tại sao cần rời rạc hóa động

Với tàu i và tài nguyên r trên tuyến cố định, đặt $t^{i,r}$ là thời điểm tàu đi vào tài nguyên. Quan hệ trước-sau trên đường đi có dạng

$$t^{i,q} \geq t^{i,r} + \ell_i^r,$$

trong khi hai tàu chia sẻ tài nguyên phải chọn một trong hai thứ tự an toàn. Nếu dựng toàn bộ miền thời gian đến khung thời gian H , số biến theo mốc thời gian có thể rất lớn.

Dynamic Discretization Discovery (khám phá rời rạc hóa động, DDD) khởi đầu bằng tập mốc thưa $\Lambda^{i,r}$. Mô hình hạn chế cho một cận dưới. Nghiệm được giải mã trên trục thời gian đầy đủ; nếu nghiệm vi phạm thời gian di chuyển hoặc khoảng cách an toàn, chỉ các mốc liên quan mới được thêm.

DDD không bắt nguồn từ SAT. Thuật toán được phát triển cho thiết kế mạng dịch vụ thời gian liên tục bằng cách giải lập các mạng mở rộng thời gian một phần và tinh chỉnh những mốc cần thiết [18]. Khi chuyển ý tưởng sang MaxSAT, điều phải bảo toàn là *tính hợp lệ của cận dưới* trong mô hình hạn chế và quy tắc tinh chỉnh hữu hạn; bộ giải bên trong có thể thay đổi. Đây là một ví dụ tiêu biểu cho quá trình tích hợp kiến trúc từ Nghiên cứu vận trù vào SAT, thay vì chỉ thay phép biểu diễn của một ràng buộc.

8.7.2. Vòng lặp

1. Giải mô hình MAXSAT hạn chế trên phép rời rạc hóa hiện tại.
2. Giải mã lịch và kiểm tra mọi quan hệ trước-sau trên đường đi, xung đột tài nguyên và giá trị độ trễ.
3. Nếu có vi phạm, thêm các mốc cần thiết và lặp lại.
4. Nếu lịch khả thi và chi phí giải mã bằng giá trị tối ưu của mô hình hạn chế, cận dưới gấp cận trên; nghiệm là tối ưu.

Đầu vào: tập mốc hữu hạn đầy đủ Λ^* , phép rời rạc hóa ban đầu $\Lambda^0 \subseteq \Lambda^*$, bộ giải mã–kiểm tra và hàm chi phí.

Đầu ra: lịch tối ưu hoặc chứng nhận vô nghiệm trong miền đầy đủ.

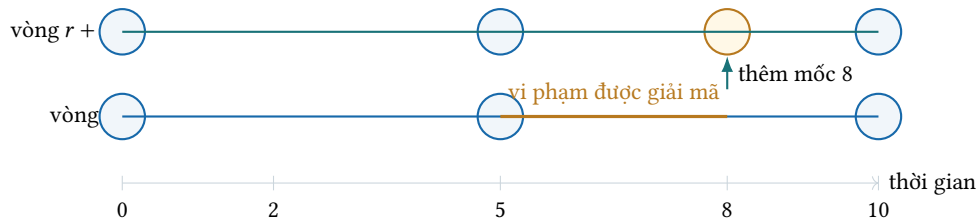
1. Với vòng r , sinh hoặc cập nhật mô hình hạn chế trên Λ^r , giải tối ưu và ghi cận dưới L_r .
2. Nếu mô hình hạn chế vô nghiệm và là một mô hình nói lỏng hợp lệ của mô hình đầy đủ, kết luận mô hình đầy đủ vô nghiệm.
3. Giải mã nghiệm. Bộ kiểm tra xác nhận một lịch đầy đủ có chi phí U_r nếu khả thi; đồng thời, khi lịch chưa khả thi hoặc $L_r < U_r$, thuật toán phải trả về một bằng chứng C_r chỉ ra phần rời rạc hóa đang thiếu cho feasibility hoặc hàm mục tiêu.
4. Nếu có lịch khả thi và $U_r = L_r$, trả lịch cùng hai cận bằng nhau.
5. Nếu chưa hội tụ, bước tinh chỉnh phải chọn ít nhất một mốc trong $\Lambda^* \setminus \Lambda^r$ được bằng chứng C_r yêu cầu; đặt $\Lambda^{r+1} \supseteq \Lambda^r$ và lặp.

Thuật toán 8.1: Vòng lặp MaxSAT–DDD có kiểm chứng

Mệnh đề 8.5 (Hội tụ hữu hạn của DDD). *Giả sử (i) tập mốc đầy đủ Λ^* hữu hạn; (ii) mỗi mô hình hạn chế cho một cận dưới hợp lệ; (iii) bộ kiểm tra đúng; và (iv) mỗi vòng chứa kết thúc thêm ít nhất một mốc mới từ Λ^* . Khi đó Thuật toán 8.1 kết thúc sau không quá $|\Lambda^* \setminus \Lambda^0| + 1$ lần giải mô hình hạn chế và chỉ trả nghiệm tối ưu hoặc kết luận vô nghiệm đúng.*

Chứng minh. Do chuỗi $\Lambda^0 \subsetneq \Lambda^1 \subsetneq \dots \subseteq \Lambda^*$ tăng nghiêm ngặt trong một tập hữu hạn, số lần tinh chỉnh bị chặn bởi $|\Lambda^* \setminus \Lambda^0|$. Nếu thuật toán dừng với $U_r = L_r$, L_r là cận dưới cho giá trị tối ưu đầy đủ còn U_r là chi phí của một nghiệm đầy đủ được bộ kiểm tra xác nhận, nên cả hai bằng giá trị tối ưu. Nếu đến Λ^* , mô hình hạn chế chính là mô hình đầy đủ; bộ giải quyết định tối ưu hoặc vô nghiệm. Kết luận vô nghiệm sớm chỉ hợp lệ dưới giả thiết nói lỏng ở (ii). $\square$

Biên hội tụ và chiến lược tinh chỉnh. Ở biên xấu nhất, DDD khôi phục toàn bộ lưới Λ^* . Mỗi bước tinh chỉnh vì thế phải thêm một mốc mới được bộ kiểm tra suy ra từ vi phạm cụ thể. Với miền thời gian



Hình 8.5.: DDD chỉ thêm mốc tại nơi nghiệm của mô hình hạn chế thiếu thông tin. Lưới không được làm dày đồng loạt trên toàn bộ khung thời gian.

liên tục, một đối chứng thời gian liên tục hoặc một định lý về tập ứng viên hữu hạn cung cấp điều kiện dừng. Ba thành phần này—mốc mới, bộ kiểm tra và tập mốc đầy đủ—xác định đồng thời tính đúng và tốc độ hội tụ.

Lan truyền quan hệ trước–sau có vai trò thu hẹp miền trước khi sinh CNF. Nếu một lựa chọn thứ tự $i < j$ kéo theo một chuỗi thời điểm tối thiểu, các mốc không đạt được bị loại ngay. AMO kết hợp tiếp tục chọn phép biểu diễn khác nhau theo kích thước miền: từng cặp cho tập rất nhỏ, tuần tự hoặc chỉ huy cho tập lớn hơn.

Trong nghiên cứu của Kieu, Nguyen, và To [52], hiệu quả thay đổi theo dạng hàm độ trễ: MAXSAT nổi bật với chi phí bậc thang và làm tròn, còn MIP cạnh tranh hơn với chi phí liên tục. Giá trị thiết kế của DDD nằm ở khả năng chỉ mở rộng miền tại nơi nghiệm hạn chế thực sự thiếu thông tin.

8.8. Chọn SAT, SMT, CP hay MIP

Không có bộ giải thống trị mọi cấu trúc. Một quy tắc thực dụng là nhìn vào phần khó nhất của mô hình:

- Chọn SAT/MaxSAT khi logic rời rạc, cửa sổ và xung đột lặp lại; có lợi từ học mệnh đề và giải tăng dần.
- Chọn SMT khi bất đẳng thức thời gian nguyên là lỗi, còn phần Boolean chỉ quyết định thứ tự hoặc chế độ [17].
- Chọn CP khi tích lũy/không chồng lấn là ràng buộc toàn cục tự nhiên và lan truyền miền có thể khai thác trực tiếp.
- Chọn MIP khi hàm mục tiêu tuyến tính mạnh và mô hình nổi lỏng cung cấp cận tốt.

Phương pháp kết hợp nên được thiết kế như một phân chia nhiệm vụ suy diễn: mô-đun nào tạo cận, mô-đun nào xử lý phép tuyển và bằng cách nào hai phía trao đổi thông tin. Việc chỉ gọi nhiều bộ giải trên cùng một mô hình không tự tạo ra một kiến trúc kết hợp.

8.8.1. Một cây quyết định theo cấu trúc

1. Nếu khung thời gian nhỏ và mỗi tác vụ có ít giá trị bắt đầu, SAT/MaxSAT theo mốc thời gian là đối chứng minh bạch.
2. Nếu khung thời gian lớn nhưng các ràng buộc trước–sau/sai phân dày, thử mã hóa thứ tự, SMT hoặc LCG trước khi mở toàn bộ trục thời gian.
3. Nếu tài nguyên tích lũy là nút thắt và có suy luận năng lượng mạnh, CP/LCG có ưu thế cấu trúc mà xung đột từng cặp khó thay thế.

4. Nếu hàm mục tiêu/mô hình nói lỏng tuyến tính tạo cận mạnh, MIP là đối chứng bắt buộc; SAT chỉ có lợi khi phép tuyển Boolean hoặc đối xứng chi phối.
5. Nếu chỉ một phần nhỏ mốc thời gian quan trọng, xét DDD/tinh chỉnh và chứng minh rõ tính hợp lệ của cận dưới.

8.9. Bài học thiết kế

1. Phân biệt biến thời điểm, biến hoạt động và biến thứ tự; viết phép liên kết trước khi tối ưu mệnh đề.
2. Tính không ngắt là một ràng buộc khối, không phải chỉ là ràng buộc đếm.
3. Với các cửa sổ chồng lấn, hiệu quả đến từ kỹ thuật chia sẻ ngưỡng hoặc bộ đếm giữa các cửa sổ.
4. Hàm mục tiêu nên giao tiếp với mô-đun khả thi qua một giao diện cận.
5. Rời rạc hóa động phù hợp khi miền lớn nhưng nghiệm chỉ cần một số ít mốc quan trọng.

Câu hỏi nghiên cứu

1. Với cùng ngữ nghĩa khối, hãy so sánh biểu diễn bằng biến bắt đầu và biểu diễn bằng điểm chuyển về số biến, số mệnh đề và khả năng lan truyền đơn vị.
2. Thiết kế một Shared Counter cho hai ALSC có cửa sổ chồng lấn nhưng cận k khác nhau.
3. Viết điều kiện dừng chính xác cho một vòng DDD khi chi phí hạn chế là cận dưới và lịch giải mã cho cận trên.

9 Đóng gói và cắt hai chiều

TÓM TẮT CHƯƠNG

Trong bài toán đóng gói, mệnh đề không xử lý trực tiếp với hình chữ nhật; mô hình làm việc với tọa độ, hướng đặt và các quan hệ tách. Chất lượng phép biểu diễn phụ thuộc vào việc thiết kế giao diện phù hợp giữa ba lớp đó.

9.1. Bản đồ các phương pháp chính xác cho đóng gói

Hai chiều tạo khó khăn khác một chiều: không chồng lấn là một phép tuyến giữa hai trục, còn tọa độ có nhiều giá trị đối xứng trong vùng trống. Văn liệu về phương pháp chính xác vì thế không xoay quanh một mô hình duy nhất. Khảo cứu của Iori và cộng sự phân biệt các mô hình theo vị trí đặt, vị trí tương đối, sinh tập/cột, quy hoạch ràng buộc và các kỹ thuật nhánh–cận chuyên biệt [45].

Đối với SAT, ba họ biểu diễn là đối chứng trực tiếp:

Chiếm chỗ

Biến cho từng ô mà vật chiếm. Không chồng lấn trở thành cục bộ nhưng số biến phụ thuộc diện tích lưới; tính liên tục của một hình cần các mệnh đề liên kết.

Vị trí đặt tuyệt đối

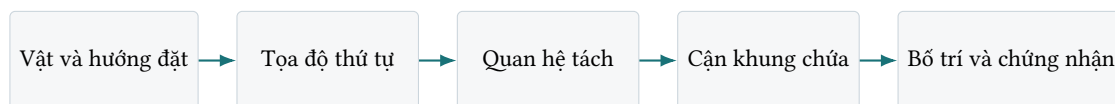
Mỗi biến chọn một vị trí khả dĩ của vật. Đồ thị xung đột nối hai vị trí chồng lấn; bài toán trở thành chọn đúng một vị trí cho mỗi vật. Các phương pháp vị trí tuyệt đối và đóng gói hoàn hảo khai thác nén tọa độ, tổng tập con và suy luận trên không gian trống [42].

Vị trí tương đối + tọa độ

Mỗi vật có tọa độ; mỗi cặp có *witness variable* (biến làm chứng) trái/phải/trên/dưới. Order Encoding nén cận tọa độ, còn mệnh đề quan hệ xử lý phép tuyến.

Ngoài SAT, quy hoạch ràng buộc dùng NoOVERLAP/diffrn, các phép chiếu tích lũy và suy luận năng lượng; MIP dùng hằng số lớn M , bất đẳng thức tuyến hoặc mẫu phủ tập. Các phương pháp chính xác cho đóng gói trực giao còn kết hợp cận, quan hệ trội và nhánh–cận chuyên dụng [23, 24].

Bảng 9.1.: Các biểu diễn chính xác cho đóng gói hai chiều.



Hình 9.1.: Lộ trình lập luận của chương: từ Vật và hướng đặt, qua Tọa độ thứ tự, Quan hệ tách và Cận khung chứa, đến Bố trí và chứng nhận.

Biểu diễn	Quy mô theo	Lan truyền tự nhiên	Phù hợp khi
Chiếm chỗ	Diện tích lưới	Xung đột ô	Khung chứa nhỏ/rất rời rạc
Vị trí đặt	Số vị trí khả dĩ	Đồ thị xung đột	Miền đã nén mạnh
Tọa độ–thứ tự	Tổng độ rộng miền	Cận và quan hệ tách	Miền nguyên vừa
CP toàn cục	Số vật + bộ lan truyền	Không chồng lấn/tích lũy	Cận hình học mạnh
MIP/phủ tập	Biến/cột vị trí	Thu giản và nhát cắt	Cận LP hữu ích

Ba trục tạo nên đóng gói SAT cho đóng gói là nén miền tọa độ, làm mạnh lan truyền không chồng lấn và tái sử dụng mệnh đề đã học giữa các cận. Kiến trúc mới được nhận diện qua thay đổi có thể đo trên ít nhất một trong ba trục này.

9.2. Mô hình hình học rời rạc

Mỗi vật i là hình chữ nhật kích thước $w_i \times h_i$. Tọa độ (x_i, y_i) chỉ góc trái dưới. Khi cho phép quay 90° , biến ρ_i xác định kích thước hiệu dụng

$$(\hat{w}_i, \hat{h}_i) = \begin{cases} (w_i, h_i), & \rho_i = 0, \\ (h_i, w_i), & \rho_i = 1. \end{cases}$$

Trong khung chứa $W \times H$, điều kiện nằm gọn là

$$0 \leq x_i \leq W - \hat{w}_i, \quad 0 \leq y_i \leq H - \hat{h}_i.$$

Hai vật i và j không giao nhau khi ít nhất một trong bốn quan hệ đúng:

$$x_i + \hat{w}_i \leq x_j, \quad x_j + \hat{w}_j \leq x_i, \quad y_i + \hat{h}_i \leq y_j, \quad y_j + \hat{h}_j \leq y_i.$$

Ta dùng các biến làm chứng l_{ij} và u_{ij} :

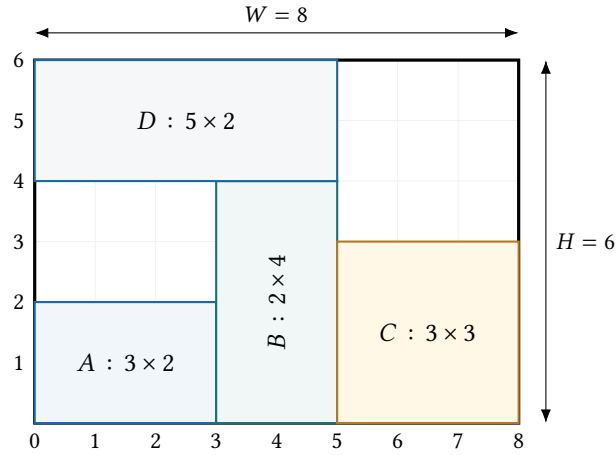
$$l_{ij} \vee l_{ji} \vee u_{ij} \vee u_{ji},$$

với quan hệ kéo theo từ từng biến làm chứng sang bất đẳng thức tương ứng. Các biến làm chứng không cần loại trừ lẫn nhau: một vật có thể vừa ở bên trái vừa ở dưới vật khác.

Tọa độ biểu diễn cạnh của hình chữ nhật, không phải tâm hay ô đã chiếm. Vì thế dấu $\leq$ cho phép hai vật chạm cạnh. Quy ước này phải được giữ nguyên trong điều kiện nằm gọn, không chồng lấn và bộ kiểm tra.

Hình 9.2 dùng một ví dụ tự dựng để minh họa cách tọa độ thứ tự và quan hệ vị trí được dùng trong SAT cho đóng gói; cách tổ chức quan hệ này tiếp nối mô hình đóng gói dải của Soh và cộng sự [73].

Trong hình 9.2, A có $(x_A, y_A) = (0, 0)$, $(w_A, h_A) = (3, 2)$; B có $(x_B, y_B) = (3, 0)$, $(w_B, h_B) = (2, 4)$.



Hình 9.2.: Một bố trí nguyên có bốn vật. Các cặp chạm cạnh vẫn hợp lệ; chẳng hạn $x_A + w_A = x_B = 3$, nên biến làm chứng l_{AB} có thể nhận giá trị đúng.

Bốn bất đẳng thức tách là

$$0 + 3 \leq 3, \quad 3 + 2 \leq 0, \quad 0 + 2 \leq 0, \quad 0 + 4 \leq 0.$$

Chỉ bất đẳng thức đầu đúng, nên $l_{AB} = 1$ là biến làm chứng tự nhiên. Nếu ép $x_B \leq 2$, miền tọa độ làm $x_A + w_A \leq x_B$ bất khả và mệnh đề liên kết buộc $\neg l_{AB}$; phép tuyển bốn quan hệ khi đó phải chọn một hướng khác hoặc chứng minh cặp hình không thể cùng một bố trí.

9.3. Order Encoding cho tọa độ

9.3.1. Ngữ nghĩa

Với e trong miền ngang, đặt

$$p_{i,e}^x = 1 \iff x_i \leq e.$$

Tương tự, $p_{i,f}^y = 1 \iff y_i \leq f$. Chuỗi phải đơn điệu:

$$p_{i,e}^x \rightarrow p_{i,e+1}^x, \quad p_{i,f}^y \rightarrow p_{i,f+1}^y.$$

Giá trị tọa độ được xác định bởi điểm chuyển đầu tiên từ sai sang đúng. Một bất đẳng thức như $x_i + a \leq x_j$ trở thành quan hệ kéo theo giữa hai ngưỡng:

$$p_{i,e}^x \text{ và } p_{j,e+a-1}^x$$

được nối theo hướng phù hợp với ngữ nghĩa $\leq$. Lợi ích là một literal ngưỡng đại diện cho cả nửa miền, thay vì liệt kê từng cặp tọa độ.

9.3.2. Hướng đặt và liên kết biểu diễn

Khi kích thước phụ thuộc ρ_i , quan hệ kéo theo

$$l_{ij} \rightarrow x_i + \hat{w}_i \leq x_j$$

được tách theo các hướng đặt của i và j . Các trường hợp không thể nằm gọn được loại ở bước tiền xử lý. Với vật vuông, đặt $\rho_i = 0$ để bỏ đối xứng quay.

Khi l_{ij} đóng vai trò biến làm chứng cho phép tuyến, mô-đun cần chiều

$$l_{ij} \rightarrow x_i + \hat{w}_i \leq x_j.$$

Chiều ngược

$$x_i + \hat{w}_i \leq x_j \rightarrow l_{ij}$$

có thể được bổ sung khi literal quan hệ cần đặc tả hai chiều. Bộ giải mã đọc trực tiếp tọa độ và hướng đặt, nhờ đó bộ kiểm tra độc lập với lựa chọn liên kết.

Mệnh đề 9.1 (Tính đúng của mô-đun không chồng lấn). *Giả sử mỗi biến làm chứng l_{ij}, u_{ij} kéo theo đúng bất đẳng thức hình học của vật, và phép tuyến bốn biến làm chứng được thỏa cho mọi cặp vật trong cùng khung chứa. Khi đó mọi bố trí giải mã từ mô hình SAT đều không giao nhau.*

Chứng minh. Với một cặp i, j , phép tuyến cho ít nhất một biến làm chứng đúng. Quan hệ kéo theo của biến làm chứng đó cho một bất đẳng thức tách theo trục x hoặc y . Vì vậy nội thất hai hình chữ nhật rời nhau. $\square$

9.4. Bài toán đóng gói (Packing problem) dải hai chiều

9.4.1. Từ tối ưu chiều cao sang kiểm tra tính khả thi

Dải có chiều rộng cố định W , còn chiều cao H cần tối thiểu hóa. Với mỗi H , dựng công thức

$$\Phi_{\text{SPP}}(H) \iff \text{tồn tại bố trí trong } W \times H.$$

Tính đơn điệu cho phép tìm kiếm trên cận:

$$\Phi_{\text{SPP}}(H) = \text{SAT} \implies \Phi_{\text{SPP}}(H') = \text{SAT} \quad \forall H' \geq H.$$

Cận dưới cơ bản là

$$LB = \max \left\{ \max_i \min(h_i, w_i), \left\lceil \frac{\sum_i w_i h_i}{W} \right\rceil \right\},$$

trong đó hạng đầu phải điều chỉnh theo hướng đặt thực sự vừa khung. Cận trên đến từ một phép đặt theo kinh nghiệm.

Chỉ cận diện tích hiếm khi đủ sát. Các phương pháp chính xác còn dùng hàm khả thi đối ngẫu, phép chiếu một chiều, tính không tương thích và tọa độ tổng tập con. Trong SAT, mọi cận dưới hợp lệ đều có hai vai trò: thu hẹp số cận phải gọi và thu hẹp miền tọa độ trước khi sinh mệnh đề. Nén tọa độ đặc biệt quan trọng: nếu có thể chứng minh một vật chỉ cần bắt đầu tại các tổng tập con khả dĩ, chuỗi thứ tự nên chạy trên tập mốc đã nén thay vì mọi số nguyên từ 0 đến $W - w_i$.

9.4.2. Ba chế độ giải

Bảng 9.2.: Ba chế độ tối ưu chiều cao bằng SAT/MaxSAT.

Chế độ	Cấu trúc	Đặc điểm
SAT độc lập	Dựng lại $\Phi(H)$ ở mỗi cận	Đơn giản, mỗi cận có một CNF riêng
SAT tăng dần	Miền lớn cố định, kích hoạt H bằng các giả định	Giữ mệnh đề đã học giữa các cận
MaxSAT có trọng số	Chiều cao được biểu diễn bằng mệnh đề mềm	Không cần vòng tìm kiếm ngoài

Biểu diễn SAT thứ tự cho đóng gói dải đã được nghiên cứu trước đó cùng đối xứng, quan hệ vị trí và tái sử dụng mệnh đề đã học [73]. Vì vậy nghiên cứu tình huống của nhóm được định vị theo hai câu hỏi kế tiếp: phép quay/đối xứng làm đổi mô hình thế nào, và SAT–MaxSAT–giải tăng dần có giữ lợi thế trên bộ kiểm chuẩn và bộ giải hiện tại hay không. Phép so sánh hợp lý phải giữ cùng các cận, miền tọa độ và bộ kiểm tra.

Trong nghiên cứu của Kieu, Le, và To [49], SAT không tăng dần với phép quay và phá vỡ đối xứng chúng nhận 26 giá trị tối ưu trên 41 bài toán; phiên bản tăng dần chúng nhận 21, còn CP chúng nhận 28. Kết quả này cho thấy giải tăng dần chỉ có lợi khi công thức cơ sở và mệnh đề đã học còn phù hợp sau khi thay cận. Với đóng gói dải, thay H có thể làm thay đổi mạnh miền tọa độ và những xung đột hữu ích.

Một bố trí tại H^* là cận trên. Công thức $\Phi(H^* - 1)$ không thỏa là chứng nhận cận dưới. Cặp SAT/UNSAT ở hai cận kề nhau mới xác định H^* là giá trị tối ưu.

9.5. Bài toán đóng gói (Packing problem) vào khung chứa hai chiều

9.5.1. Hai kiến trúc

Bài toán dùng nhiều khung chứa $W \times H$ đồng nhất và tối thiểu hóa số khung. Có hai kiến trúc chính.

Xếp chồng. Ghép k khung thành một dải $W \times kH$. Tọa độ y_i đồng thời xác định khung. Kiến trúc này tái sử dụng đóng gói dải nhưng phải ngăn vật vượt qua đường biên qH .

Không xếp chồng. Dùng biến $a_{ib} = 1$ khi vật i nằm trong khung b , và tọa độ cục bộ trong từng khung. Ta có

$$\text{ExactlyOne}(a_{i1}, \dots, a_{ik}).$$

Điều kiện không chồng lấn của cặp i, j trong khung b được gác bởi:

$$(a_{ib} \wedge a_{jb}) \rightarrow (l_{ij} \vee l_{ji} \vee u_{ij} \vee u_{ji}).$$

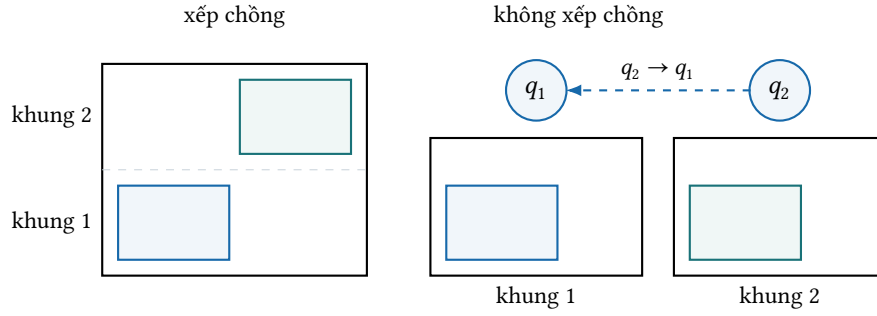
Biến sử dụng q_b nối với phép gán bằng $a_{ib} \rightarrow q_b$. Đối xứng giữa các khung được giảm bằng

$$q_{b+1} \rightarrow q_b.$$

Các khung đã dùng tạo thành một tiền tố, vì thế hàm mục tiêu $\sum_b q_b$ đúng bằng số khung. Có thể thêm phép gán chuẩn cho một số vật đầu, nhưng mọi ràng buộc đối xứng phải dựa trên phép hoán vị các khung đồng nhất, không dựa trên một bố trí kinh nghiệm cụ thể.

Trong mô hình theo vị trí đặt, một cách khác là xây đồ thị xung đột cho mọi cặp vị trí–khung rồi thêm ràng buộc đúng một theo vật. Cách này tránh biến quan hệ nhưng có thể tạo số vị trí rất lớn. Mô hình tọa độ không xếp chồng làm ngược lại: số biến tọa độ nhỏ hơn toàn bộ tập vị trí, đổi lại mỗi cặp vật–khung cần một phép tuyển có điều kiện. Hai mô hình là hai phía của phép phân tích nhân tử; đánh giá thực nghiệm nên báo cả số vị trí bị loại ở tiền xử lý và số quan hệ từng cặp còn lại.

Nghiên cứu của Kieu, Hoang, và To [47] so sánh các mô hình SAT với MIP và CP trên cùng họ bài toán kiểm chuẩn. Kết quả nhấn mạnh hai yếu tố hơn là một phép biểu diễn đơn lẻ: biến thứ tự giúp biểu diễn bất đẳng thức tọa độ, còn phá đối xứng cho khung/bản sao quyết định đáng kể kích thước không gian tìm kiếm.



Hình 9.3.: Hai kiến trúc cho đóng gói vào khung: xếp chồng dùng một trục y toàn cục; không xếp chồng dùng phép gán và tọa độ cục bộ, kèm tiền tố sử dụng.

9.6. Bài toán cắt tẩm một kích thước

9.6.1. Mô hình theo số tờ

Cho tấm vật liệu kích thước $W \times H$, nhiều loại vật t , mỗi loại có nhu cầu d_t . Bài toán tối thiểu hóa số tấm. Nếu tạo từng bản sao riêng, phép gán và không chồng lẫn giống bài toán đóng gói (packing problem) vào khung. Nếu gộp theo loại, cần thêm ràng buộc đếm để bảo đảm đúng nhu cầu.

Với một cận K , công thức khả thi hỏi liệu có thể đáp ứng đủ nhu cầu trên K tấm. Hàm mục tiêu là

$$\min \{K \mid \Phi_{\text{CSP}}(K) \text{ là SAT}\}.$$

Đối xứng xuất hiện ở hai tầng: các tấm đồng nhất và các bản sao cùng loại. Một cách phá vỡ đối xứng bền vững là buộc biến sử dụng tấm theo tiền tố và buộc chỉ số tấm của các bản sao cùng loại không giảm.

9.6.2. SAT và MaxSAT

Tìm kiếm SAT thay K ở vòng ngoài. MAXSAT bộ phận có thể giữ tất cả các tấm ứng viên, đặt mệnh đề mềm $\neg q_b$ và tối thiểu $\sum_b q_b$. Hai cách chia sẻ điều kiện nằm gọn, hướng đặt và không chồng lẫn; chúng chỉ khác ở lớp hàm mục tiêu.

Kieu, Hoang, và To [48] cho thấy phép biểu diễn gọn với tọa độ thứ tự và đối xứng theo bản sao/tấm đạt kết quả cạnh tranh trên bộ bài toán cắt tẩm một kích thước. Điểm chuyển giao sang các bài toán đóng gói (packing problem) khác là chuỗi mô-đun

Assignment + OrderCoordinate + ConditionalNonOverlap + UsageCounter.

9.7. Giảm miền và đối xứng

Trước khi sinh CNF, bốn phép rút gọn thường có hiệu quả trực tiếp:

1. loại hướng đặt không vừa khung;
2. thu hẹp $x_i \leq W - \hat{w}_i$ và $y_i \leq H - \hat{h}_i$;
3. loại một hướng tách nếu các cận chứng minh hướng đó bất khả thi;
4. chuẩn hóa vật/bản sao và khung chứa theo một thứ tự cố định.

Mỗi ràng buộc phá vỡ đối xứng được gán với một nhóm đối xứng cụ thể:

- $\rho_i = 0$ cho vật vuông: đối xứng hướng đặt;
- $q_{b+1} \rightarrow q_b$: hoán vị khung/tấm;

- thứ tự bản sao cùng loại: hoán vị các bản sao đồng nhất;
- neo một vật lớn vào nửa khung chứa: phản xạ/tịnh tiến, nếu phép biến đổi giữ miền và hàm mục tiêu.

Các nguyên tắc tổng quát về đối xứng được trình bày trong Gent, Petrie, và Puget [38] và Walsh [88]; phép biểu diễn SAT cổ điển cho bài toán đóng gói (packing problem) khung 2D Packing có rotation có thể xem ở Soh, Tamura, và Banbara [74].

9.8. Tách ảnh hưởng của phép biểu diễn, cận và quan hệ trội

Ma trận thiết kế ghi riêng ảnh hưởng của các thành phần:

- cận dưới theo diện tích, phép chiếu và đồ thị con đầy đủ của quan hệ không tương thích;
- loại hướng đặt và vị trí bị trội;
- cố định theo vật lớn hoặc thứ tự vật đồng nhất;
- nén tọa độ;
- khởi động âm/cận trên từ phương pháp kinh nghiệm.

Ma trận thiết kế tách phần giảm mệnh đề do lọc miền khỏi phần giảm do chuyển sang CNF. Phép so sánh CP, MIP và SAT chia sẻ cách chuẩn hóa bài toán và cận đầu vào; cận và quan hệ trội được ghi thành mô-đun riêng [23, 45].

9.9. Một thư viện mô-đun cho hình học

OrderCoordinate cung cấp chuỗi ngưỡng và phép giải mã duy nhất. **Orientation** trả về kích thước hiệu dụng. **Containment** siết miền theo khung chứa đang chọn. **RelativePosition** tạo biến làm chứng trái/phải/trên/dưới. **ConditionalNonOverlap** gác phép tuyển bằng phép gán. **UsageCounter** biến số khung chứa được dùng thành hàm mục tiêu.

Thứ tự kết hợp đi từ hướng đặt đến điều kiện nằm gọn, vị trí tương đối, không chồng lấn được gác bởi phép gán và bộ đếm sử dụng. Bộ kiểm tra đọc vật, hướng đặt, tọa độ và khung chứa, độc lập với biến phụ.

9.10. Bài học thiết kế

1. Chọn một quy ước tọa độ và duy trì quy ước này nhất quán trong toàn bộ phép biểu diễn.
2. Dùng biến làm chứng cho phép tuyển hình học; không buộc biến làm chứng trở thành quan hệ đầy đủ nếu phép giải mã không cần.
3. Tách mô-đun khả thi khỏi hàm mục tiêu số khung/tấm/chiều cao.
4. Phá vỡ đối xứng theo đúng phép hoán vị hoặc phản xạ của mô hình.
5. Chứng nhận tối ưu gồm bố trí hợp lệ và một cận dưới tương ứng.

Câu hỏi nghiên cứu

1. Dẫn xuất các mệnh đề thứ tự cho quan hệ kéo theo $l_{ij} \rightarrow x_i + \hat{w}_i \leq x_j$ ở một hướng đặt cố định.

2. So sánh số biến của xếp chồng và không xếp chồng trên n vật, k khung.
3. Thiết kế ràng buộc phá vỡ đối xứng cho ba bản sao cùng loại và chứng minh mỗi quỹ đạo giữ một đại diện.

10 Bandwidth, antibandwidth và tô đa màu

TÓM TẮT CHƯƠNG

Các bài toán trong chương đều nói về khoảng cách giữa nhãn, nhưng không chia sẻ một ngữ nghĩa: antibandwidth gán một hoán vị, tô màu theo bandwidth gán một màu, còn tô đa màu gán một tập màu có thứ tự.

10.1. Bố trí đồ thị, gán nhãn và tô màu

Từ “bandwidth” được dùng cho nhiều bài toán có công thức gần nhau nhưng lịch sử thuật toán khác nhau. Bandwidth đồ thị cổ điển tối thiểu hóa khoảng cách lớn nhất trên cạnh của một hoán vị đỉnh. Antibandwidth tối đa hóa khoảng cách nhỏ nhất. Tô màu theo bandwidth bỏ tính song ánh và cho trọng số khoảng cách trên cạnh; tô đa màu cho mỗi đỉnh nhiều tần số. Văn liệu gán nhãn đồ thị rộng hơn còn chứa hàng trăm quy tắc gán số khác nhau [36].

Ba truyền thống giải liên quan trực tiếp:

- **cận lý thuyết đồ thị:** bậc, số ổn định, số sắc, *clique* (đồ thị con đầy đủ) và cấu trúc lớp đồ thị;
- **phương pháp kinh nghiệm/siêu kinh nghiệm:** tìm kiếm cục bộ, *tabu search* (tìm kiếm kiêng kỵ), lân cận biến đổi và thứ tự kiến tạo để tìm nghiệm đương nhiệm cho cận trên/cận dưới;
- **phương pháp chính xác:** quy hoạch động, nhánh–cận, MIP, CP và SAT theo cận.

SAT giải hiệu quả bài toán quyết định “có phép gán nhãn tại cận k không” và kết hợp tự nhiên với các cận lý thuyết đồ thị. Các cận thu hẹp khoảng tìm kiếm, tạo các giả định đầu tiên tốt hơn và đôi khi trực tiếp chứng nhận tối ưu. Với BCP/BMCP, khảo cứu tô màu đỉnh và các biến thể cung cấp mô hình, cận theo đồ thị con đầy đủ và nguồn gốc bộ kiểm chuẩn cần dùng làm đối chứng [57].

Bảng 10.1.: Bốn họ bài toán khoảng cách nhãn.

Họ	Phép gán	Cận miền tự nhiên	Đối chứng chính xác
Bandwidth	Hoán vị	Cận bố trí/bậc	DP, B&B, MIP
Antibandwidth	Hoán vị	Cận ổn định/số sắc	MIP, CP, SAT
BCP	Một màu/đỉnh	Cận đồ thị con đầy đủ và tô màu	MIP, CP, SAT
BMCP	Nhiều màu/đỉnh	Cận ô màu/nhu cầu	MIP, tìm kiếm cục bộ, SAT



Hình 10.1.: Lộ trình lập luận của chương: từ Hoán vị hay tô màu, qua Cửa sổ/thứ tự/khối, Khoảng cách và Tìm cận, đến Gán nhãn và chứng nhận.

10.2. Phân loại trước khi biểu diễn

Bảng 10.2.: Phân biệt ngữ nghĩa của ABP, CABP, No-hole Anti- k -labeling, BCP và BMCP.

Bài toán	Nghiem	Ràng buộc	Mục tiêu
Antibandwidth	Hoán vị $1, \dots, n$	$ f(u) - f(v) \geq k$	Tối đa k
Antibandwidth tuần hoàn	Hoán vị trên vòng	$D_c(f(u), f(v)) \geq k$	Tối đa k
No-hole labeling	Anti-k- Ánh xạ toàn ánh $\{1, \dots, k\}$ ($k < n$)	$ \psi(u) - \psi(v) \geq w$ và $\sum_i x_{ij} \geq 1$	Tối đa w
Tô màu theo bandwidth	Một màu mỗi đỉnh	$ c(u) - c(v) \geq d_{uv}$	Tối thiểu độ trải
Tô đa màu theo bandwidth	$w(u)$ màu mỗi đỉnh	Khoảng cách nội/ngoại đỉnh	Tối thiểu màu lớn nhất

Với ABP và CABP, nhãn là một hoán vị nên ma trận gán cần ExactlyOne theo cả hàng và cột. Với BCP, hai đỉnh không kề nhau có thể dùng cùng màu; chỉ hàng cần ExactlyOne. Với BMCP, một đỉnh có nhiều ô màu; thứ tự các ô là một phần của mô hình.

Hai phép biểu diễn có cùng biến $x_{v,\ell}$ vẫn có thể mô tả hai bài toán khác nhau. Hãy viết phép giải mã và ràng buộc đếm theo hàng/cột trước khi thêm các ràng buộc khoảng cách.

10.3. Antibandwidth và cửa sổ staircase

10.3.1. Cận và phương pháp chính xác ngoài SAT

Với đồ thị liên thông, các cận trên ABP có thể lấy từ số đỉnh/cạnh, bậc, số ổn định $\alpha(G)$ và số sắc $\chi(G)$. Chẳng hạn

$$AB(G) \leq \alpha(G), \quad AB(G) \leq \left\lfloor \frac{n-1}{\chi(G)-1} \right\rfloor.$$

Tính chính xác α, χ cũng khó, nhưng một cận trên cho α hoặc cận dưới cho χ vẫn có thể tạo cận hợp lệ. Các mô hình MIP, nhánh-cắt và CP đã cải thiện đáng kể chúng nhận trên bộ Harwell-Boeing [71]. Do đó một nghiên cứu SAT trên cùng bộ kiểm chuẩn cần báo cùng cận đầu vào và so cả giá trị nghiệm lẫn khoảng cách chứng minh, không chỉ số bài toán có mô hình.

10.3.2. Tính khả thi ở cận k

Cho đồ thị $G = (V, E)$, $|V| = n$. Biến

$$x_{v,\ell} = 1 \iff f(v) = \ell$$

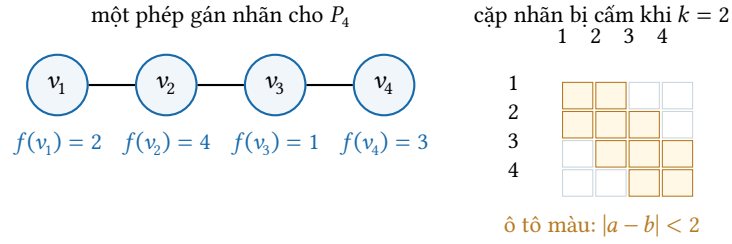
mô tả một hoán vị:

$$\text{ExactlyOne}(x_{v,1}, \dots, x_{v,n}) \quad \forall v,$$

$$\text{ExactlyOne}(x_{1,\ell}, \dots, x_{n,\ell}) \quad \forall \ell.$$

Với một cận k , mỗi cạnh (u, v) cấm hai nhãn cách nhau dưới k :

$$\neg x_{u,a} \vee \neg x_{v,b}, \quad |a - b| < k.$$



Hình 10.2.: Ở trái, nhãn $(2, 4, 1, 3)$ cho P_4 đạt antibandwidth 2. Ở phải, mỗi ô được tô biểu diễn một cặp nhãn bị cấm trên một cạnh.

Một cách nhìn khác cố định vị trí a của u . Khi đó v không được nằm trong cửa sổ

$$[a - k + 1, a + k - 1] \cap [1, n].$$

Khi a trượt, các cửa sổ chồng lấn tạo một họ AMO bậc thang. Nếu biểu diễn từng cửa sổ độc lập, phần lớn bộ đếm bị lặp.

10.3.3. Thang Sequential Counter

SCL chia dãy biến thành các khối, xây Sequential Counter trong mỗi khối rồi nối trạng thái ở biên. Ý tưởng cốt lõi là biến phụ mang nghĩa số đếm tiền tố:

$$s_{i,j} = 1 \iff \sum_{q=1}^i x_q \geq j.$$

Các cửa sổ chia sẻ tiền tố/hậu tố thay vì dựng lại bộ đếm. Với trường hợp $k = 1$, đây là phép biểu diễn cho AMO bậc thang; với AMK, bộ đếm giữ nhiều mức $j = 1, \dots, k$.

Mệnh đề 10.1 (Tái sử dụng qua biên khối). *Nếu bộ đếm trong mỗi khối đúng với mọi tiền tố cục bộ và bộ nối bảo toàn tổng tích lũy khi đi qua biên, thì ngưỡng của mọi cửa sổ ghép từ hậu tố khối trái và tiền tố khối phải đúng với số literal đúng trong cửa sổ đó.*

Chứng minh. Tổng của cửa sổ bằng tổng trên hai phần rời nhau. Bộ đếm cục bộ biểu diễn đúng hai tổng; bộ nối liệt kê các cặp ngưỡng có tổng vượt cận. Vì vậy một phép gán vi phạm ràng buộc đếm kích hoạt ít nhất một mệnh đề nối, còn một phép gán hợp lệ mở rộng được bằng các giá trị đếm tiền tố thật. $\square$

Truong, Kieu, và To [80] cho thấy SCL giảm cả biến phụ và mệnh đề so với nhiều phép biểu diễn không chia sẻ trên họ AMO bậc thang. Hướng AMK được phát triển trong Truong, Kieu, và To [81]; Duplex của Fazekas và cộng sự [35] là một đối chiếu quan trọng theo hướng BDD hai chiều. Một cách nhìn rộng hơn về biến phụ có cấu trúc được thảo luận trong Haberlandt, Green, và Heule [40].

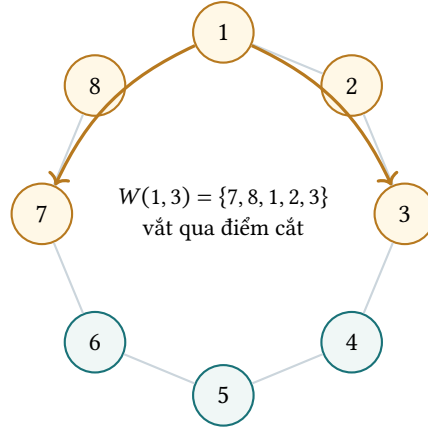
SCL bổ sung cho các mô hình MIP/CP một cơ chế nén các khoảng xung đột trong cnf. Cận mạnh và SCL tương hỗ trực tiếp: k quyết định chiều rộng cửa sổ cần biểu diễn, còn Shared Counter giảm phần biểu diễn lặp giữa các cửa sổ đó.

10.4. Antibandwidth tuần hoàn

10.4.1. Khoảng cách vòng

Trong CABP, nhãn nằm trên vòng C_n :

$$D_c(a, b) = \min\{|a - b|, n - |a - b|\}.$$



Hình 10.3.: Cửa sổ cấm trên vòng C_8 với tâm 1 và $k = 3$. Hai nhánh 7, 8 và 2, 3 phải được nối khi tuyến tính hóa vòng.

Với cận k , một cạnh cấm cửa sổ vòng bán kính $k - 1$. Cửa sổ có thể vắt qua điểm cắt nên dùng chỉ số môđun:

$$W(a, k) = \{a - k + 1, \dots, a + k - 1\} \pmod{n}.$$

Xung đột theo nhãn chính xác vẫn đúng nhưng số mệnh đề lớn. Cửa sổ vòng tạo AMO tuần hoàn, có thể biểu diễn bằng cách cắt vòng tại một vị trí neo rồi nối hai đầu bằng bộ nối.

10.4.2. Đối xứng của vòng

CABP bất biến dưới phép quay và phản xạ của nhãn. Ta có thể cố định một đỉnh đại diện tại nhãn 1, rồi chọn hướng chuẩn cho một đỉnh thứ hai. Đây là phá đối xứng ở cấp hoán vị; điều kiện này độc lập với SCL dùng cho các cửa sổ khoảng cách.

Tìm nghiệm tối ưu theo cận tăng:

$$F(k) = \text{SAT} \implies F(k - 1) = \text{SAT}, \quad F(k) = \text{UNSAT} \implies F(k + 1) = \text{UNSAT}.$$

Một nghiệm tại k^* và UNSAT tại $k^* + 1$ chứng nhận tối ưu.

Nghiên cứu của Truong và To [84] xây dựng các phép biểu diễn SAT cho CABP và khai thác cửa sổ tuần hoàn cùng đối xứng của vòng. Một tập công thức khả thi được chuẩn bị cho SAT Competition 2026 [83]; về mặt thiết kế, giá trị của bộ này là tách bài toán tối ưu thành chuỗi bài toán SAT/UNSAT có cận k rõ ràng.

10.5. Bài toán No-hole Anti- k -labeling và nén khoảng cấm bằng biến tiền tố

10.5.1. Bài toán gán nhãn phi song ánh và điều kiện toàn ánh

Trong quản lý phổ tần số và tài nguyên mạng không dây, số lượng kênh truyền dẫn khả thi k thường nhỏ hơn đáng kể so với số lượng trạm phát sóng $n = |V|$ ($k < n$) do các giới hạn về mặt pháp lý và kỹ thuật. Bối cảnh này dẫn tới bài toán *Anti- k -labeling* [39], một mở rộng phi song ánh tự nhiên của bài toán Antibandwidth cổ điển (nơi $k = n$). Khi $k < n$, phép gán nhãn không còn là một hoán vị mà trở thành một phép phân hoạch các đỉnh vào các lớp nhãn.

Biến thể *No-hole Anti-k-labeling* [41, 89] yêu cầu hàm gán nhãn $\psi : V \rightarrow \{1, \dots, k\}$ phải là một phép ánh xạ *toàn ánh* (*surjective mapping*), tức là mọi nhãn trong tập $\{1, \dots, k\}$ đều phải được gán cho ít nhất một đỉnh (không có nhãn trống/lỗ hổng). Mục tiêu của bài toán là tìm hàm gán nhãn ψ sao cho khoảng cách giữa hai đỉnh kề nhau là lớn nhất:

$$\lambda_k^{nh}(G) = \max_{\psi} \min_{(u,v) \in E} |\psi(u) - \psi(v)|. \quad (10.1)$$

Mô hình quy hoạch tuyến tính số nguyên (ILP) cho bài toán quyết định “liệu có tồn tại phép gán nhãn toàn ánh thỏa mãn khoảng cách tối thiểu w không” sử dụng các biến nhị phân $x_{ij} = 1 \iff \psi(i) = j$:

$$\sum_{j=1}^k x_{ij} = 1 \quad \forall i \in \{1, \dots, n\}, \quad (10.2)$$

$$\sum_{i=1}^n x_{ij} \geq 1 \quad \forall j \in \{1, \dots, k\}, \quad (10.3)$$

$$x_{uj} + \sum_{j'=\max(1,j-w+1)}^{\min(k,j+w-1)} x_{vj'} \leq 1 \quad \forall (u, v) \in E, j \in \{1, \dots, k\}. \quad (10.4)$$

Ràng buộc (10.2) bảo đảm mỗi đỉnh nhận đúng một nhãn; Ràng buộc (10.3) bảo đảm tính toàn ánh (*no-hole*); và Ràng buộc (10.4) cấm hai đỉnh kề nhau nhận các nhãn thuộc khoảng cấm $(j - w, j + w)$.

10.5.2. Tủ bùng nổ mệnh đề từng cặp đến biểu diễn khoảng bằng biến tiền tố

Chuyển trực tiếp Ràng buộc khoảng cách (10.4) sang CNF bằng phép biểu diễn từng cặp (Direct Encoding - DE) sinh ra họ mệnh đề xung đột:

$$\neg x_{ui} \vee \neg x_{vj} \quad \forall (u, v) \in E, i \in \{1, \dots, k\}, j \in [\max(1, i - w + 1), \min(k, i + w - 1)]. \quad (10.5)$$

Số lượng mệnh đề của DE phát triển theo cấp số $O(|E| \cdot k \cdot w)$. Khi khoảng cách mục tiêu w tăng lên, số mệnh đề khoảng cách bùng nổ mạnh mẽ, gây tắc nghẽn bộ nhớ và làm suy giảm hiệu năng của bộ giải SAT.

Để giải quyết triệt để nút thắt này, phép biểu diễn PSE (*Proposed SAT Encoding*) [41] giới thiệu các biến phụ tiền tố $l_{ij} \in \{0, 1\}$ với ngữ nghĩa:

$$l_{ij} = 1 \iff \psi(i) \leq j \iff \exists k' \leq j : x_{ik'} = 1.$$

Tính đơn điệu tự nhiên của biến tiền tố đòi hỏi $l_{ij} \rightarrow l_{i(j+1)}$ cho mọi $j < k$. Hệ thống mệnh đề liên kết biến gán gốc x_{ij} và biến tiền tố l_{ij} được định nghĩa bởi:

$$x_{ij} \rightarrow l_{ij} \quad \forall i \in \{1, \dots, n\}, j \in \{1, \dots, k\}, \quad (10.6)$$

$$l_{ij} \rightarrow l_{i(j+1)} \quad \forall i \in \{1, \dots, n\}, j \in \{1, \dots, k-1\}, \quad (10.7)$$

$$\neg x_{i1} \rightarrow \neg l_{i1} \quad \forall i \in \{1, \dots, n\}, \quad (10.8)$$

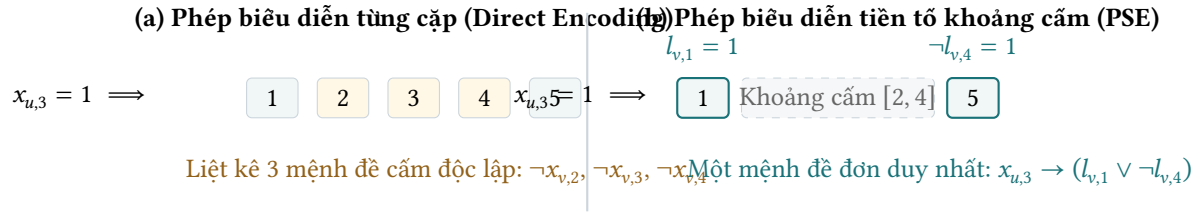
$$(\neg x_{ij} \wedge \neg l_{i(j-1)}) \rightarrow \neg l_{ij} \quad \forall i \in \{1, \dots, n\}, j \in \{2, \dots, k\}, \quad (10.9)$$

$$x_{ij} \rightarrow \neg l_{i(j-1)} \quad \forall i \in \{1, \dots, n\}, j \in \{2, \dots, k\}. \quad (10.10)$$

Nhờ cấu trúc biến tiền tố, toàn bộ khoảng cấm $(j - w, j + w)$ giữa hai đỉnh kề nhau $(u, v) \in E$ được nén lại chỉ trong **đúng một mệnh đề kéo theo đơn duy nhất**:

$$x_{uj} \rightarrow (l_{v, \max(1, j-w)} \vee \neg l_{v, \min(k, j+w-1)}) \quad \forall (u, v) \in E, j \in \{1, \dots, k\}. \quad (10.11)$$

Mệnh đề (10.11) khẳng định rằng: nếu đỉnh u nhận nhãn j , thì đỉnh kề v bắt buộc phải nhận nhãn nhỏ hơn/bằng $j - w$ (thỏa mãn literal $l_{v, \max(1, j-w)}$) hoặc phải nhận nhãn lớn hơn $j + w - 1$ (thỏa mãn literal $\neg l_{v, \min(k, j+w-1)}$). Điều này tự động loại bỏ tất cả các nhãn nằm trong khoảng cấm $[j - w + 1, j + w - 1]$ mà không cần liệt kê từng cặp nhãn vi phạm.



Hình 10.4.: So sánh cơ chế nén khoảng cấm khi $k = 5, w = 2$: (a) Phép biểu diễn từng cặp liệt kê $O(w)$ mệnh đề; (b) Phép biểu diễn tiền tố PSE loại toàn bộ khoảng cấm bằng một mệnh đề kéo theo duy nhất.

Mệnh đề 10.2 (Độ phức tạp và tính đúng đắn của PSE). *Phép biểu diễn PSE biểu diễn chính xác bài toán No-hole Anti- k -labeling. Tổng số biến nhị phân là $2nk$, tổng số mệnh đề khoảng cách là $2|E|k$, và các mệnh đề liên kết là $n(4k - 2)$. Phép biểu diễn hoàn toàn triệt tiêu sự phụ thuộc của số mệnh đề vào tham số khoảng cách mục tiêu w .*

Chứng minh. Mỗi đỉnh i có k biến gán gốc x_{ij} và k biến tiền tố l_{ij} , tổng cộng $2nk$ biến. Các Mệnh đề (10.6)–(10.10) bảo đảm ngữ nghĩa $l_{ij} = \sum_{k' \leq j} x_{ik'}$ và ép buộc đỉnh i nhận đúng một nhãn mà không cần thêm các mệnh đề EXACTLYONE bổ sung. Mệnh đề khoảng cách (10.11) chỉ tạo ra 2 mệnh đề cho mỗi cặp cạnh định hướng và nhãn j , cho tổng số $2|E|k$ mệnh đề. $\square$

10.5.3. Phá vỡ đối xứng phản xạ và kết quả thực nghiệm

Không gian nghiệm của bài toán No-hole Anti- k -labeling chứa đối xứng phản xạ: nếu $\psi(v)$ là một nghiệm hợp lệ, thì nghiệm đối xứng $\psi'(v) = k + 1 - \psi(v)$ cũng hợp lệ và giữ nguyên khoảng cách $|\psi'(u) - \psi'(v)| = |\psi(u) - \psi(v)|$ [41]. Để phá vỡ đối xứng này và thu hẹp không gian tìm kiếm, phép biểu diễn cố định một đỉnh chốt p (thường chọn đỉnh có chỉ số 0) thuộc nửa dưới miền nhãn:

$$\bigwedge_{i=\lfloor k/2 \rfloor + 1}^k \neg x_{pi}. \quad (10.12)$$

Trên 23 bộ dữ liệu chuẩn Harwell–Boeing [41]:

- Phép biểu diễn PSE kết hợp với bộ giải song song PAINLESS giải thành công **100% các bài toán**, vượt trội hoàn toàn so với mô hình DE (bị bùng nổ mệnh đề và không tìm được nghiệm khả thi trên 3 bài toán lớn U, V, X trong thời gian 1800 giây).
- So với các công cụ thương mại hàng đầu đại diện cho MIP (Gurobi, CPLEX) và CP (CPLEX CP)—vốn hoàn toàn thất bại không tìm được bất kỳ nghiệm khả thi nào trên 8 bài toán quy mô lớn từ M đến X—phép biểu diễn PSE cải thiện trung bình **19%** chất lượng nghiệm mục tiêu và giảm trên **82%** thời gian chạy trên các bài toán giải được.

10.6. Tô màu theo bandwidth

10.6.1. Quan hệ với tô màu đỉnh

Khi $d_{uv} = 1$ trên mọi cạnh, BCP trở về tô màu đỉnh theo độ trải. Khi các trọng số lớn hơn, một đồ thị con đầy đủ không chỉ yêu cầu các màu khác nhau mà còn yêu cầu một phép đóng gói một chiều của các khoảng cách. Vì vậy kích thước đồ thị con đầy đủ vẫn cho cận dưới nhưng không mô tả đủ phép tách có trọng số. Các mô hình MIP theo phép gán, đại diện hoặc phủ tập trong tô màu đỉnh là điểm khởi đầu; BCP phải bổ sung các xung đột có xét khoảng cách [57].

Từ góc nhìn biểu diễn, biến trực tiếp mở rộng mô hình gán; biến thứ tự nén một dãy giá trị xung đột thành hai biên; biến khối tạo các đại diện khoảng. Sáu họ dưới đây khảo sát có hệ thống trực biểu diễn đó và kết nối với các cận lý thuyết đồ thị đã nêu.

10.6.2. Mô hình và sáu họ biểu diễn

BCP cho đồ thị có trọng số cạnh $d_{uv} > 0$. Mỗi đỉnh nhận màu $c(u) \in [1, H]$ và

$$|c(u) - c(v)| \geq d_{uv} \quad \forall uv \in E.$$

Mục tiêu là tối thiểu

$$\chi_B(G) = \min_c \max_{u \in V} c(u).$$

Nghiên cứu hệ thống của Nguyen, Kieu, và To [62] tổ chức không gian thiết kế thành sáu họ:

Bảng 10.3.: Sáu họ biểu diễn cho tô màu theo bandwidth.

Họ	Biến	Cơ chế khoảng cách
1G	$g_{u,j} \iff c(u) \geq j$	Một chuỗi thứ tự hướng lớn hơn
1L	$l_{u,j} \iff c(u) \leq j$	Một chuỗi thứ tự hướng nhỏ hơn
2G	Giá trị đúng $x_{u,j}$ và ngưỡng $g_{u,j}$	Giá trị chính xác kích hoạt hai biên khoảng cấm
2L	Giá trị đúng $x_{u,j}$ và ngưỡng $l_{u,j}$	Nhu 2G với hướng thứ tự đảo
X	Biến khối	Chia sẻ cấu trúc khoảng cách theo khối
X_a	Khối và phép trừ phụ	Tái sử dụng hiệu/khoảng qua biến phụ

10.6.3. Khoảng cấm bằng hai biên

Giả sử $x_{u,a} = 1$. Khi đó màu của v phải nằm ngoài

$$[a - d_{uv} + 1, a + d_{uv} - 1].$$

Với ngữ nghĩa $g_{v,t} = 1 \iff c(v) \geq t$, điều kiện được viết bằng một mệnh đề biên:

$$\neg x_{u,a} \vee \neg g_{v,a-d_{uv}+1} \vee g_{v,a+d_{uv}}.$$

Hai literal cuối lần lượt nói $c(v) < a - d_{uv} + 1$ hoặc $c(v) \geq a + d_{uv}$. Số mệnh đề không còn phụ thuộc vào độ rộng khoảng cấm; mỗi giá trị chính xác tạo một mệnh đề cho mỗi hàng xóm.

Biểu diễn theo khối chia miền màu thành các đoạn và dùng biến phụ để tóm tắt quan hệ giữa các đoạn. CNF có thể lớn hơn Order Encoding, nhưng lan truyền trên bài toán khó có thể mạnh hơn vì bộ giải nhìn thấy các cấu trúc khoảng được chia sẻ.

Cho $H = 7$, $x_{u,4} = 1$ và $d_{uv} = 2$. Đỉnh v không được nhận

$$[4 - 2 + 1, 4 + 2 - 1] = [3, 5].$$

Với $g_{v,t} \iff c(v) \geq t$, chỉ cần mệnh đề

$$\neg x_{u,4} \vee \neg g_{v,3} \vee g_{v,6}.$$

Khi $x_{u,4} = 1$, hai literal còn lại nói đúng hai miền hợp lệ: $c(v) \leq 2$ hoặc $c(v) \geq 6$. Direct Encoding

từng cặp phải cấm riêng $(4, 3), (4, 4), (4, 5)$; biểu diễn theo biên giữ số mệnh đề không đổi khi khoảng cấm rộng hơn, miễn tập cấm vẫn là một khoảng.

Trên tập GEOM và MS-CAP của Nguyen, Kieu, và To [62], cấu hình theo khối giải được các bài toán khó mà Order Encoding không hoàn tất trong cùng giới hạn. Giải tăng dần và đối xứng không có hiệu ứng đồng đều: lợi ích phụ thuộc vào họ biểu diễn. Vì vậy thí nghiệm phân tích thành phần (ablation study) nên dùng đơn vị phép biểu diễn $\times$ đặc trưng, không chỉ so trung bình từng đặc trưng.

10.7. Tô đa màu theo bandwidth

10.7.1. Mô hình ô màu và giảm miền

BMCP gán cho đỉnh u một tập $w(u)$ màu. Viết chúng thành các ô có thứ tự

$$c(u)_1 < c(u)_2 < \dots < c(u)_{w(u)}.$$

Hai ô cùng đỉnh cách nhau ít nhất $d(u, u)$; hai ô ở hai đỉnh kề nhau cách ít nhất $d(u, v)$. Với cận màu lớn nhất k , miền ô được siết:

$$LB_{u,i} = 1 + (i - 1)d(u, u),$$

$$UB_{u,i} = k - (w(u) - i)d(u, u).$$

Thứ tự ô loại đối xứng hoán vị màu bên trong một đỉnh và đồng thời tạo lan truyền cận.

Mô hình ô là một phép mở rộng miền, trong khi mô hình tần số là một mô hình tập. Hai cách có đối xứng và cận dưới khác nhau. Trong mô hình ô, thứ tự nội bộ loại $w(u)!$ hoán vị; trong mô hình tập, đối xứng đó không được tạo ra nhưng ràng buộc “chọn đúng $w(u)$ màu” cần ràng buộc đếm. So sánh FE/SBE/OBE vì vậy phải bao gồm cả số trạng thái gán trước các mệnh đề khoảng cách.

10.7.2. FE, SBE và OBE

Flat Encoding (biểu diễn phẳng, FE). Biến $x_{u,m}$ cho biết màu m được dùng tại u . Ràng buộc đếm bảo đảm đúng $w(u)$ màu; xung đột từng cặp cấm các cặp quá gần.

Slot-Based Encoding (biểu diễn theo ô màu, SBE). Biến $x_{u,i,m}$ nói ô i bằng m . Mỗi ô có ràng buộc đúng một và các cặp ô vi phạm khoảng cách bị cấm trực tiếp.

Order-Based Encoding (biểu diễn theo thứ tự, OBE). Thêm $g_{u,i,m} = 1 \iff c(u)_i \geq m$. Giá trị chính xác tại một ô kích hoạt hai biên khoảng cấm của ô ở đỉnh khác:

$$x_{u,i,m} \rightarrow (g_{v,j,m+d(u,v)} \vee \neg g_{v,j,m-d(u,v)+1}).$$

Số mệnh đề xung đột giảm một bậc theo miền màu so với SBE từng cặp.

Mệnh đề 10.3 (Đúng dẫn của OBE). *Nếu liên kết giá trị chính xác–thứ tự xác định cùng một giá trị cho mỗi ô, chuỗi thứ tự đơn điệu và mọi giá trị chính xác kích hoạt mệnh đề khoảng cấm ở trên, thì mọi mô hình OBE giải mã thành một phép tô đa màu theo bandwidth hợp lệ.*

Chứng minh. Ràng buộc đúng một cho mỗi ô một màu; các mệnh đề cận và thứ tự giữ các ô trong miền và đúng thứ tự. Với mỗi cặp ô bị ràng buộc, giá trị chính xác m của ô thứ nhất buộc ô thứ hai nằm dưới $m - d + 1$ hoặc từ $m + d$ trở lên. Vì vậy khoảng cách tuyệt đối ít nhất d . $\square$

Tìm kiếm tăng dần từ trên xuống bắt đầu từ một cận trên khả thi. Sau khi tìm mô hình có độ trái s , bộ giải cấm các màu từ s trở lên và tiếp tục. Lần UNSAT đầu tiên tại $s - 1$, cùng mô hình tại s , chúng nhận tối ưu. Nguyen và To [63] dùng cơ chế này để chứng minh tối ưu cho BMCP.

10.8. Một trừu tượng chung: giá trị chính xác và khoảng cấm

ABP, BCP và BMCP gặp cùng một mẫu:

1. một biến chính xác chọn giá trị a ;
2. lựa chọn đó tạo một khoảng giá trị bị cấm cho đối tượng khác;
3. biến thứ tự nén khoảng thành hai literal biên.

Khác biệt nằm ở nguồn của khoảng và số lựa chọn trong phép gán. ABP có ràng buộc khác nhau toàn cục; BCP không có; BMCP có nhiều ô trên một đỉnh. Vì vậy mô-đun `ForbiddenInterval` có thể dùng lại, còn `AssignmentSemantics` phải được tham số hóa theo bài toán.

10.9. Quy trình chính xác kết hợp

Một quy trình có giá trị tham khảo cho cả bốn bài toán là:

1. dùng cận lý thuyết đồ thị và phương pháp kinh nghiệm để tạo khoảng $[LB, UB]$;
2. chọn mô hình đúng ngữ nghĩa: hoán vị, một màu hoặc nhiều ô;
3. lọc miền bằng cận cục bộ hoặc đồ thị con đầy đủ trước khi tạo biến;
4. chọn Direct Encoding/thứ tự/khối theo hình dạng tập cấm;
5. giải theo quy trình cận đơn điệu và lưu nghiệm làm chứng;
6. tại cận cuối, kiểm tra phép gán nhãn miền và chứng minh UNSAT độc lập.

Kiến trúc này đặt SAT trong một khung giải chính xác thay vì coi bộ giải SAT là toàn bộ thuật toán. Cấu trúc này cũng làm rõ vị trí đóng góp của một nghiên cứu mới: cận, mô hình, phép biểu diễn, quy trình tìm kiếm hay chứng nhận.

Bài học trung tâm. Chọn Direct Encoding, thứ tự hay theo khối không chỉ là chọn kích thước CNF. Biến trực tiếp làm phép giải mã và xung đột cục bộ rõ; biến thứ tự nén bất đẳng thức và khoảng; biến khối tạo điểm chia sẻ mới cho lan truyền. Một kiến trúc tốt thường kết hợp cả ba ở những lớp khác nhau.

Câu hỏi nghiên cứu

1. Với ABP trên đường P_4 , dựng $F(k)$ tại $k = 2$ và chỉ rõ exactly-one theo hàng lẫn cột.
2. Dẫn xuất mệnh đề biên BCP cho ba trường hợp: khoảng nằm trong miền, chạm biên trái và chạm biên phải.

3. Thiết kế thí nghiệm phân tích thành phần (ablation study) 2×2 cho giải tăng dần và đối xứng trên hai họ biểu diễn BCP.

11 Gán nhãn và phân bổ tần số

TÓM TẮT CHƯƠNG

Radio labeling (gán nhãn radio), k -safe labeling (gán nhãn an toàn k) và frequency assignment (phân bổ tần số) cùng chia sẻ một mô-đun: một lựa chọn chính xác tạo ra forbidden interval (khoảng cấm) trên miền của đối tượng khác. Sự khác nhau nằm ở cách tạo khoảng cấm và cách định nghĩa hàm mục tiêu.

11.1. Hai văn liệu nguồn: gán nhãn đồ thị và phân bổ tần số

Các bài toán trong chương đi vào SAT từ hai văn liệu khác nhau. Gán nhãn đồ thị nghiên cứu các ánh xạ số trên đỉnh/cạnh dưới nhiều quy tắc khoảng cách; gán nhãn radio là một nhánh trong hệ phân loại rất rộng đó [36]. Phân bổ tần số bắt nguồn từ mô hình nhiễu vô tuyến: mỗi máy phát có miền, ràng buộc phân cách, đôi khi có giá trị gán trước, phân cực, nhiều loại nhiễu và nhiều hàm mục tiêu.

Khảo cứu của Aardal và cộng sự phân loại FAP theo miền, mô hình nhiễu và hàm mục tiêu, đồng thời tổng hợp MIP, cận dưới lý thuyết đồ thị, tối ưu ràng buộc và phương pháp kinh nghiệm [1]. Từ góc nhìn SAT, điểm quan trọng là không đồng nhất ba hàm mục tiêu:

- **độ trải nhỏ nhất**: giảm hiệu giữa tần số lớn nhất và nhỏ nhất;
- **số giá trị nhỏ nhất**: giảm số tần số phân biệt được dùng;
- **nhiều/chi phí nhỏ nhất**: cho phép vi phạm hoặc mức nhiễu có trọng số.

Chúng có thể dùng cùng các ràng buộc khả thi nhưng cần biến hàm mục tiêu và lập luận cận dưới khác nhau.

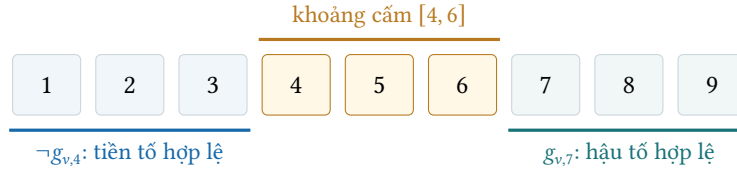
Bảng 11.1.: Ba lớp bài toán gán nhãn và tần số.

Lớp	Miền	Ràng buộc	Hàm mục tiêu điển hình
Gán nhãn radio	Liên tục $[1, B]$	Từ khoảng cách đồ thị	Độ trải/nhiều lớn nhất nhỏ nhất
Đơn ánh an toàn k	Liên tục $[1, B]$	Khác nhau + khoảng cách cạnh	Nhiều lớn nhất nhỏ nhất
FAP	Rời rạc theo máy phát	Phân cách nhiễu	Độ trải, số giá trị hoặc chi phí

Các công trình của nhóm trong chương được dùng để minh họa ba phép chuyển: xung đột từng cặp sang biên khoảng; miền liên tục sang miền rời rạc; và cận độ trải sang bộ đếm giá trị phân biệt. Ba phép chuyển này kết nối hệ phân loại FAP và gán nhãn đồ thị với các mô-đun SAT có thể dùng lại.



Hình 11.1.: Lộ trình lập luận của chương: từ Giá trị chính xác, qua Chuỗi thứ tự, Khoảng cấm và Độ trải hoặc số giá trị, đến Nghiệm chứng và cận dưới.



Hình 11.2.: Một khoảng cấm liên tục được nén thành hai literal biên: $\neg g_{v,4} \vee g_{v,7}$.

11.2. Bỏ đề khoảng cấm

Cho $x_{u,a} = 1$ khi đối tượng u nhận giá trị a . Với đối tượng v , đặt

$$g_{v,t} = 1 \iff f(v) \geq t.$$

Giả sử lựa chọn a cấm khoảng $[L(a), U(a)]$ cho v . Điều kiện hợp lệ là

$$f(v) < L(a) \quad \vee \quad f(v) > U(a).$$

Trên miền nguyên liên tục, ràng buộc trở thành

$$\neg x_{u,a} \vee \neg g_{v,L(a)} \vee g_{v,U(a)+1}. \quad (11.1)$$

Bổ đề 11.1 (Mệnh đề biên). *Giả sử chuỗi thứ tự $g_{v,t}$ đúng với ngữ nghĩa $f(v) \geq t$. Khi $x_{u,a} = 1$, mệnh đề (11.1) được thỏa khi và chỉ khi giá trị của v nằm ngoài $[L(a), U(a)]$.*

Chứng minh. Nếu $x_{u,a} = 1$, literal đầu sai. Literal $\neg g_{v,L(a)}$ đúng khi $f(v) < L(a)$; literal $g_{v,U(a)+1}$ đúng khi $f(v) \geq U(a) + 1$. Phép tuyển vì thế tương đương với điều kiện nằm ngoài khoảng. $\square$

Trên miền rời rạc D_v , không thể dùng trực tiếp chỉ số L, U . Ta thay chúng bằng hai biên theo giá trị:

$$\lambda_v(a) = \min \{q \mid D_{v,q} \geq L(a)\},$$

$$\rho_v(a) = \min \{q \mid D_{v,q} > U(a)\}.$$

Các literal biên sau đó dùng $\lambda_v(a)$ và $\rho_v(a)$ trong chuỗi thứ tự theo chỉ số miền.

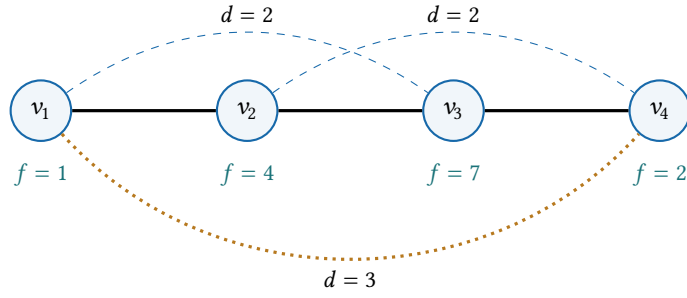
11.2.1. Quan hệ với Support Encoding

Trong Direct Encoding/hỗ trợ, lựa chọn $x_{u,a}$ yêu cầu v nhận một trong các giá trị còn hỗ trợ:

$$\neg x_{u,a} \vee \bigvee_{b \in D_v: b \notin [L(a), U(a)]} x_{v,b}.$$

Mệnh đề biên nén phép tuyển này thành hai ngưỡng khi tập hỗ trợ là hợp của một tiền tố và một hậu tố. Đây không phải một mẹo riêng của radio/FAP; đó là phép phân tích mệnh đề hỗ trợ dựa trên tính lồi của tập cấm. Nếu tập cấm có nhiều khoảng rời nhau, cần nhiều cặp biên hoặc một biểu diễn BDD/MDD khác.

Phép so sánh lan truyền theo dõi đường suy luận của từng biểu diễn. Mệnh đề hỗ trợ trực tiếp loại giá trị chính xác; mệnh đề biên suy qua chuỗi thứ tự và liên kết giá trị chính xác–thứ tự. Hai phép biểu diễn đạt cùng mức cắt tĩa khi đặc tả của các chuỗi đó cung cấp đủ các bước suy luận [7, 37].



Hình 11.3.: Một phép gán nhãn radio 3 khả thi cho P_4 . Hai đỉnh kề nhau cần chênh ít nhất 3, cặp cách hai cạnh cần 2, và hai đầu mút cần 1.

ExactValue chọn a ; OrderChain biểu diễn miền của v ; ForbiddenInterval chỉ cần một hàm trả về hai biên. Gán nhãn radio, gán nhãn an toàn k và MO-FAP khác nhau ở hàm này, không khác nhau ở logic của định lý 11.1.

11.3. Radio k -labeling: gán nhãn radio k

11.3.1. Mô hình

Cho đồ thị liên thông $G = (V, E)$, khoảng cách đường đi ngắn nhất $d(u, v)$ và tham số k . Phép gán nhãn radio k là ánh xạ $f : V \rightarrow \mathbb{Z}_{>0}$ thỏa

$$|f(u) - f(v)| \geq \Delta_{uv}, \quad \Delta_{uv} = k + 1 - d(u, v),$$

cho mọi cặp có $\Delta_{uv} > 0$. Mục tiêu là tối thiểu màu lớn nhất

$$\min \max_{v \in V} f(v).$$

Trong bài toán gán nhãn radio chuẩn, người ta thường xét $k = \text{diam}(G)$.

Khoảng cách đồ thị tạo một đồ thị ràng buộc dày hơn đồ thị gốc: mọi cặp có $d(u, v) \leq k$ đều bị ràng buộc, với mức phân cách giảm khi khoảng cách đồ thị tăng. Bước tìm đường đi ngắn nhất cho mọi cặp vì thế không chỉ dành cho bộ kiểm tra xác định toàn bộ cấu trúc liên kết của phép biểu diễn. Các cặp có $\Delta_{uv} \leq 0$ được bỏ, còn các đỉnh trong cùng quỹ đạo có thể hỗ trợ phá đối xứng.

11.3.2. SAT trực tiếp

Với một cận trên B , đặt

$$x_{v,\ell} = 1 \iff f(v) = \ell, \quad \ell \in [1, B].$$

Mỗi đỉnh nhận đúng một nhãn. Với cặp u, v , mọi cặp nhãn (a, b) có $|a - b| < \Delta_{uv}$ tạo xung đột

$$\neg x_{u,a} \vee \neg x_{v,b}.$$

Direct Encoding có phép giải mã đơn giản nhưng số xung đột tăng theo độ rộng khoảng cấm.

11.3.3. SAT thú tị

Thêm

$$g_{v,\ell} = 1 \iff f(v) \geq \ell.$$

Biến chính xác được xác định tại điểm chuyển:

$$x_{v,\ell} \iff g_{v,\ell} \wedge \neg g_{v,\ell+1}.$$

Nếu u nhận a , v không được nhận nhãn trong $[a - \Delta_{uv} + 1, a + \Delta_{uv} - 1]$. Mệnh đề biên là

$$\neg x_{u,a} \vee \neg g_{v,a-\Delta_{uv}+1} \vee g_{v,a+\Delta_{uv}},$$

sau khi bỏ các literal nằm ngoài miền. Số mệnh đề khoảng cách trở thành tuyến tính theo B cho mỗi cặp đỉnh, thay vì phụ thuộc thêm vào Δ_{uv} .

11.3.4. Tối thiểu hóa độ trải theo cách tăng dần

Bộ giải được dựng một lần ở cận trên B . Sau khi tìm mô hình có giá trị lớn nhất β , thêm

$$\neg g_{v,\beta} \quad \forall v$$

để buộc mọi nhãn không vượt quá $\beta - 1$. Vòng lặp nhảy theo nghiệm dương nhiệm thật, không cần giảm từng đơn vị. Nếu lần gọi ở $\beta - 1$ trả UNSAT, mô hình tại β cùng cận dưới này chứng nhận tối ưu.

Đối xứng có thể khai thác theo tự đẳng cấu của đồ thị. Với chu trình C_n , sau khi dịch nhãn nhỏ nhất về 1, phép quay đưa đỉnh mang nhãn 1 về một đỉnh neo cố định. Quy tắc neo như vậy phải dựa trên quỹ đạo của đồ thị.

Trên 146 bài toán từ chín họ đồ thị, Thanh, Van, và To [78] báo 38 số radio mới, khớp hoặc cải thiện BKS trên 130 bài toán và chứng nhận 109 giá trị tối ưu khi kết hợp các khung SAT với ILP. SAT thú vị đặc biệt hiệu quả ở những họ có đường kính tăng theo kích thước; ILP bổ sung tốt ở các họ có đường kính bị chặn.

11.4. Gán nhãn an toàn k chính xác

11.4.1. Mô hình được sử dụng

Trong mô hình của Thanh và cộng sự [77], nhãn

$$L : V \rightarrow \{1, \dots, B\}$$

thỏa hai điều kiện:

$$\begin{aligned} L_u &\neq L_v & \forall u \neq v, \\ |L_u - L_v| &\geq k & \forall uv \in E. \end{aligned}$$

Mục tiêu là tối thiểu $\max_v L_v$. Điều kiện đầu là tính đơn ánh toàn cục: mỗi nhãn được dùng nhiều nhất một lần.

Chương giữ đúng mô hình đơn ánh ở trên. Vì vậy, khi $k = 1$, bài toán vẫn là gán nhãn đơn ánh; đây không phải bài toán tô màu đồ thị chuẩn. Biến thể không đơn ánh là một mô hình khác.

11.4.2. Direct, forward order và reverse order encoding

Direct Encoding (Direct Encoding, DE). $K_{v,\ell}$ nói $L_v = \ell$. Ràng buộc đúng một theo đỉnh và AMO theo nhãn thực hiện phép gán đơn ánh. Mỗi cạnh cấm các cặp nhãn cách dưới k .

Forward Order Encoding (Order Encoding xuôi, FE). $X_{v,j} = 1 \iff L_v \geq j$. Giá trị chính xác xuất hiện ở điểm chuyển và kích hoạt hai biên của khoảng $[j - k + 1, j + k - 1]$.

Reverse Order Encoding (Order Encoding ngược, RE). $X_{v,j} = 1 \iff L_v \leq j$. Ngữ nghĩa đối xứng với FE; hướng của chuỗi thứ tự và các literal cận được đảo.

FE và RE có cùng tập phép gán nhãn nhưng có thể dẫn bộ giải đi theo các nhánh khác nhau. Đối xứng phản xạ

$$L'_v = S + 1 - L_v$$

giữ mọi khoảng cách và tính đơn ánh trên miền $[1, S]$. Có thể chọn một đỉnh đại diện và đặt biến đại diện ở nửa dưới miền để giữ một đại diện của cặp phản xạ.

Trên 16 đồ thị Erdős–Rényi của Thanh và cộng sự [77], FE và RE tìm nghiệm cho toàn bộ tập và chúng nhận 13 giá trị tối ưu; DE tìm nghiệm cho 15 và chúng nhận 8. Kết quả cho thấy biên thứ tự phù hợp với ràng buộc khoảng cách, còn hướng FE/RE tạo khác biệt nhỏ hơn lựa chọn trực tiếp/thứ tự.

11.5. Phân bố tần số với số giá trị nhỏ nhất

11.5.1. Miền rời rạc và hàm mục tiêu

Mỗi máy phát v có miền tần số đã sắp

$$D_v = \{D_{v,1} < D_{v,2} < \dots < D_{v,m_v}\}.$$

Các cạnh mang khoảng cách nhiều; một số máy phát có giá trị gán trước. Hàm mục tiêu không phải tần số lớn nhất mà là số giá trị phân biệt được dùng:

$$\min|\{f_v \mid v \in V\}|.$$

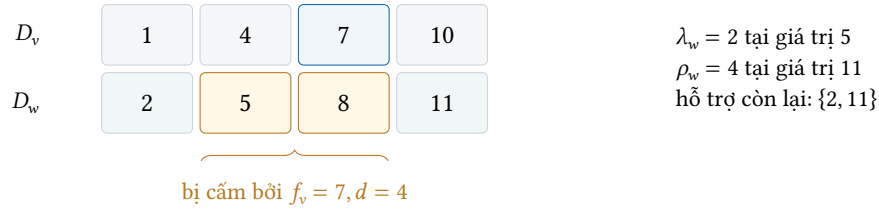
Do miền có lỗ và khác nhau giữa các máy phát, khoảng cấm phải được tính theo giá trị thực của tần số rời ánh xạ về chỉ số miền.

Trong hệ phân loại FAP, đây là mô hình số giá trị nhỏ nhất: hai nghiệm dùng cùng số giá trị phân biệt có cùng giá trị mục tiêu dù độ trải khác nhau. Do đó chuỗi thứ tự theo *chỉ số miền cục bộ* chỉ phục vụ tính khả thi; biến A_a theo *giá trị toàn cục* mới phục vụ hàm mục tiêu. Trộn hai loại thứ tự này là một lỗi mô hình phổ biến.

Xét ba máy phát có miền chứa $\{1, 4, 7\}$, và không có cạnh nhiều giữa máy phát thứ hai với thứ ba. Hai phương án

$$f = (1, 4, 7), \quad f' = (1, 7, 7)$$

đều có tần số lớn nhất 7, nhưng lần lượt dùng 3 và 2 giá trị phân biệt. Tối thiểu hóa độ trải xem chúng ngang nhau; FAP số giá trị nhỏ nhất ưu tiên f' . Vì vậy biến hàm mục tiêu phải là A_1, A_4, A_7 , không phải chỉ một ngưỡng “tần số lớn nhất không vượt 7”.



Hình 11.4.: Ảnh xạ khoảng giá trị $[4, 10]$ vào hai biên của miền rời rạc $D_w = \{2, 5, 8, 11\}$.

11.5.2. Tiền xử lý nhất quán cung

Với cặp (v, w) , một giá trị $a \in D_v$ phải có ít nhất một giá trị hỗ trợ $b \in D_w$ thỏa khoảng cách yêu cầu. Nếu không có, a bị loại. Quá trình lặp đến *fixpoint* (điểm bất động, tức không còn giá trị nào bị loại):

$$D_v \leftarrow \{a \in D_v \mid \exists b \in D_w : |a - b| \geq d_{vw}\}$$

cho mọi cung ràng buộc. Tiền xử lý này không thay giá trị tối ưu vì chỉ loại giá trị không thể xuất hiện trong bất kỳ nghiệm khả thi nào.

Đây là AC-3 trên các ràng buộc phân cách nhị phân. Khả năng suy diễn lan truyền của AC-3 bị giới hạn ở hỗ trợ từng cạnh: một giá trị có thể có hỗ trợ riêng trên mọi đỉnh kề nhưng không có hỗ trợ đồng thời toàn cục. Sau tiền xử lý, học mệnh đề SAT xử lý các tổ hợp xung đột còn lại. Vì vậy nên báo kích thước miền tại điểm bất động và thời gian AC riêng; không gán toàn bộ mức giảm thời gian chạy cho POSE.

11.5.3. DSE và POSE

Biểu diễn SAT trực tiếp dùng $x_{v,q} = 1 \iff f_v = D_{v,q}$, ràng buộc đúng một theo máy phát và xung đột từng cặp cho các cặp giá trị vi phạm.

Biểu diễn SAT thứ tự bộ phận thêm

$$g_{v,q} = 1 \iff \text{chỉ số tần số của } v \text{ không nhỏ hơn } q.$$

Nếu $x_{v,q} = 1$, tần số $D_{v,q}$ tạo khoảng cấm $[D_{v,q} - d + 1, D_{v,q} + d - 1]$ trong miền D_w . Hai biên λ_w, ρ_w của miền rời rạc được đưa vào mệnh đề biên

$$\neg x_{v,q} \vee \neg g_{w,\lambda_w(q)} \vee g_{w,\rho_w(q)}.$$

POSE vì thể loại sự phụ thuộc vào số giá trị nằm bên trong khoảng cấm.

Với các miền trong hình 11.4, chọn giá trị thứ ba của D_v , tức $x_{v,3} = 1$ và $f_v = 7$. Khoảng cách $d = 4$ cấm $[4, 10]$ ở D_w . Mệnh đề POSE là

$$\neg x_{v,3} \vee \neg g_{w,2} \vee g_{w,4}.$$

Khi $x_{v,3} = 1$, $\neg g_{w,2}$ cho phép chỉ số 1 (tần số 2); $g_{w,4}$ cho phép chỉ số 4 (tần số 11). Hai giá trị 5, 8 nằm giữa bị loại mà không cần hai mệnh đề xung đột riêng.

11.5.4. Đếm số tần số phân biệt

Với mỗi giá trị tần số toàn cục a , tạo biến

$$A_a = 1 \iff \exists v : f_v = a.$$

Liên kết $x_{v,q} \rightarrow A_{D_{v,q}}$ bảo đảm mọi tần số được dùng đều được đánh dấu. Hàm mục tiêu trở thành một ràng buộc đếm:

$$\min \sum_a A_a.$$

Sequential Counter tăng dần được dùng một lần đến cận trên ban đầu. Sau khi mô hình dùng q tần số, lần gọi kế tiếp kích hoạt cận $q - 1$ bằng các giả định. Lần UNSAT tại $q - 1$ cùng mô hình tại q chứng nhận tối ưu.

Trên 35 bài toán CALMA/DUTtest hoặc được mô hình hóa lại, Dao, Dang, và To [28] báo cấu hình POSE kết hợp bộ đếm tăng dần chứng nhận trạng thái khả thi hoặc vô nghiệm của toàn bộ tập thử. Nhất quán cung giảm mạnh tổng thời gian của cấu hình này. Kết quả cho thấy vai trò bổ sung của ba mô-đun: lọc miền, nén khoảng cấm và đếm giá trị phân biệt.

11.6. So sánh theo cấu trúc

Bảng 11.2.: So sánh cấu trúc Radio k , k -Safe và MO-FAP.

Đặc trưng	Radio k	k -Safe	MO-FAP
Miền	$[1, B]$ liên tục	$[1, B]$ liên tục	D_v rời rạc
Khoảng cấm	Từ khoảng cách đồ thị	k trên cạnh	Từ nhiễu
Cặp bị ràng buộc	Mọi cặp có $\Delta > 0$	Cạnh và tính đơn ánh	Cạnh ràng buộc
Hàm mục tiêu	Nhãn lớn nhất nhỏ nhất	Nhãn lớn nhất nhỏ nhất	Số giá trị khác nhau nhỏ nhất
Lớp mục tiêu	Cận độ trái	Cận độ trái	Bộ đếm giá trị phân biệt

Ba bài toán chia sẻ mô-đun khả thi ở mức khoảng, nhưng lớp hàm mục tiêu khác nhau. Radio và k -safe thay cận trên của nhãn. MO-FAP giữ các giá trị số cố định và tối thiểu số giá trị được kích hoạt. Đây là lý do không nên đồng nhất “số giá trị nhỏ nhất” với “độ trái nhỏ nhất”.

11.7. Đối chứng ngoài SAT và cận dưới

Đối với độ trái gán nhãn, MIP dùng biến nhãn nguyên và hướng nhị phân cho trị tuyệt đối; CP dùng ràng buộc khác nhau và khoảng cách; các phép dựng lý thuyết đồ thị cho cận hoặc nghiệm trên từng họ đồ thị. Đối với FAP, MIP/sinh cột, tối ưu ràng buộc, tìm kiếm cục bộ và *tabu search* (tìm kiếm kiêng kỵ) là các đối chứng đã được phát triển lâu dài [1]. Một đánh giá SAT có giá trị nên phân biệt:

1. bài toán mà SAT cải thiện nghiệm đương nhiệm;
2. bài toán mà SAT đóng khoảng cách bằng chứng nhận UNSAT;
3. bài toán mà phương pháp khác cung cấp cận dưới/cận trên đầu vào;
4. bài toán vô nghiệm do miền/giá trị gán trước, có thể phát hiện trước bước tối ưu.

Với số giá trị nhỏ nhất, cận dưới cơ bản có thể đến từ cấu trúc đồ thị con đầy đủ hoặc cấu trúc nhiễu hoặc số giá trị bắt buộc bởi các giá trị gán trước; cận trên đến từ phương án tần số kinh nghiệm. Bộ đếm giá trị phân biệt SAT tiếp nhận hai cận và lần lượt đóng khoảng $[LB, UB]$.

11.8. Bộ kiểm tra và chứng nhận

Bộ kiểm tra khả thi đọc trực tiếp nghiệm:

- radio: kiểm miền, đường đi ngắn nhất cho mọi cặp và mọi $\Delta_{uv} > 0$;
- k -safe: kiểm miền, tính đơn ánh và phân cách trên cạnh;
- MO-FAP: kiểm $f_v \in D_v$, giá trị gán trước, mọi ràng buộc và số tần số phân biệt.

Bộ kiểm tra không đọc các biến thứ tự phụ. Điều này giúp bộ kiểm tra giữ kích thước nhỏ gọn và độc lập với phép biểu diễn.

Chứng nhận tối ưu gồm một nghiệm làm chứng tại cận trên và bằng chứng cận dưới: UNSAT ở độ trái/số giá trị kề trước, hoặc một cận dưới toán học bằng cận trên. Với các giả định tăng dần, có thể hiện thực hóa (reification) các giả định cuối thành mệnh đề đơn vị để tạo một CNF độc lập cho bộ kiểm tra chứng minh.

1. Khoảng cấm là trừu tượng chung của cả ba bài toán.
2. Trên miền liên tục, khoảng dùng hai ngưỡng trực tiếp; trên miền rời rạc, phải tìm biên theo giá trị.
3. Tối thiểu hóa độ trái và tối thiểu hóa số giá trị phân biệt cần hai mô đun hàm mục tiêu khác nhau.
4. Biến trực tiếp phục vụ lựa chọn chính xác và giải mã; biến thứ tự nén khoảng; Sequential Counter đếm số giá trị được dùng.

Câu hỏi nghiên cứu

1. Chứng minh định lý 11.1 khi khoảng chạm một biên của miền.
2. Với $D_v = \{10, 30, 40\}$ và khoảng $[15, 35]$, xác định hai literal biên của POSE.

12 Kết luận và hướng nghiên cứu

TÓM TẮT CHƯƠNG

Chương kết nối các kết quả kỹ thuật thành một luận điểm chung: một phép biểu diễn SAT tốt là một giao diện có ngữ nghĩa, được chọn theo cấu trúc bài toán và được đánh giá bằng một chuỗi bằng chứng có thể kiểm tra. Từ đó, chương tổng hợp miền hiệu quả của từng họ phương pháp và xây dựng chương trình nghiên cứu tiếp theo.

12.1. Những kết luận bền vững

Các họ bộ biểu diễn tạo thành một biên Pareto (Pareto frontier) thay vì một thứ hạng duy nhất. Quy trình lựa chọn có thể dừng lại gồm bốn lớp.

Lớp ngữ nghĩa. Trước khi đếm biến và mệnh đề, phải viết đặc tả cho biến chính, biến phụ, bộ giải mã và hàm mục tiêu. Tính đúng, tính đầy đủ theo phép chiếu và khả năng dùng literal phụ ở mô-đun khác là ba yêu cầu khác nhau. Một biến Tseitin chỉ được xem như ngưỡng, biến làm chứng hay trạng thái khi các mệnh đề thật sự bảo đảm ngữ nghĩa đó.

Lớp cấu trúc. Phép biểu diễn nên khớp với hình học của ràng buộc. Cửa sổ chồng lấn gợi ý trạng thái chia sẻ; miền hữu hạn có thứ tự gợi ý literal thứ tự; tổng Boolean gợi ý bộ đếm, Totalizer, mạng đếm hoặc BDD; phép tuyến hình học gợi ý biến quan hệ. Đối xứng nên được loại trừ hết ở cách chọn biến, sau đó mới bằng ràng buộc có phép ánh xạ bảo toàn nghiệm.

Lớp giải. Đánh giá ở lớp giải kết hợp số mệnh đề với lan truyền đơn vị, đường truyền qua biến phụ, khả năng giữ mệnh đề đã học khi thay cận, chi phí của bộ giải mã/bộ kiểm tra và tương tác với tiền xử lý. Cùng một phép biểu diễn khả thi có thể cư xử khác nhau dưới SAT độc lập, SAT tăng dần, MaxSAT hay một kiến trúc kết hợp.

Lớp bằng chứng. Phát biểu về tính đúng cần chứng minh hoặc kiểm tra vết cận trên miền nhỏ; phát biểu về kích thước cần công thức và bộ đếm từ gói tái lập; phát biểu về hiệu năng cần quy trình, đối chứng, giới hạn thời gian và dữ liệu từng lần chạy. Tối ưu chỉ được chứng nhận khi cận trên từ một nghiệm khả thi gặp cận dưới hợp lệ, thường là một kết quả UNSAT ở cận kế tiếp.

12.2. Tổng hợp các nghiên cứu tình huống

Bảng 12.1 tổng hợp thiết kế đánh giá, kết quả và cơ chế biểu diễn nổi bật của các nghiên cứu tình huống.

Bảng 12.1.: Tổng hợp các nghiên cứu tình huống.

Mô-đun/bài toán	Thiết kế đánh giá	Kết quả chính
SCL/SCAMO và Anti-Bandwidth	Phân tích kích thước SCAMO; 24 đồ thị kiểm chuẩn	Thang chia sẽ giảm phần bộ đếm lắp trên AMO bậc thang và cạnh tranh trong cấu hình Anti-Bandwidth đã thử; độ chồng lấn khoảng là đặc trưng quyết định [80].
NSC/Xếp nhóm chơi golf	66 bài toán; NSC, ADDER, BDD, CARD, BINOMIAL	NSC giải 56/66; ADDER, BDD và CARD cùng 54/66, BINOMIAL 32/66. Kết quả định vị NSC như một lựa chọn hiệu quả cho ràng buộc đúng K trong mô hình bài toán xếp nhóm chơi golf của nguồn [50].
ALSCE/Lập lịch điều dưỡng	28 bài toán, đến 120 điều dưỡng và 24 tuần	ALSCE giải toàn bộ tập; kết quả hỗ trợ việc chia sẽ thanh ghi cho các cửa sổ chuỗi trong mô hình này [82].
SAT/Đóng gói dài	41 bài toán; SAT thường, SAT tăng dần và CP	Số giá trị tối ưu được chứng nhận lần lượt là 26, 21 và 28. SAT tăng dần phát huy lợi thế khi thay cận giữ được phần lớn miền và mệnh đề đã học [49].
Gán nhãn radio	146 bài toán từ chín họ đồ thị; các khung SAT và ILP	Công trình báo 109 giá trị tối ưu được chứng nhận, 38 số radio mới, và khớp hoặc cải thiện BKS trên 130 bài toán; SAT và ILP có vai trò bổ sung theo cấu trúc đồ thị [78].
Gán nhãn an toàn k	16 đồ thị Erdős–Rényi; FE, RE và DE	FE và RE tìm nghiệm cho 16/16, chứng nhận 13 giá trị tối ưu; DE tìm nghiệm cho 15 và chứng nhận 8. Biên thứ tự phù hợp hơn Direct Encoding trong tập thử này [77].
POSE/MO-FAP	35 bài toán CALMA/DUTtest hoặc được mô hình hóa lại	POSE kết hợp bộ đếm tăng dần quyết định toàn bộ tập thử; kết quả hỗ trợ kiến trúc lọc miền, nén khoảng cấm và đếm số tần số khác nhau [28].

12.3. Miền hiệu quả và điểm chuyên thiết kế

Shared Counter và Ladder AMK. Shared Counter được ưu tiên khi các cửa sổ chồng lấn mạnh, cận k nhỏ và mỗi cửa sổ cắt ít khối. Khi số biên tăng, bộ đếm ghép thay bộ nối hai ngôi; với các tập gần rời nhau, một phép biểu diễn cục bộ thường gọn hơn.

NSC. NSC có kích thước $O(nk)$ và tiết kiệm vùng trạng thái tam giác bất khả. Cấu trúc này phù hợp nhất khi vùng này chiếm tỷ lệ đáng kể và bài toán dùng một cận cố định. Totalizer/mạng đếm phù hợp với nhiều cận tăng dần; khi k gần n , đếm literal bù chuyển bài toán về ngưỡng $n - k$ nhỏ hơn.

Phá vỡ đối xứng. Mỗi ràng buộc phá vỡ đối xứng được suy từ một nhóm tự đẳng cấu bảo toàn miền, ràng buộc và hàm mục tiêu. Sức chứa, miền quay và cách hàm mục tiêu đọc chỉ số quyết định những hoán vị nào thuộc nhóm này. Mức phá vỡ đối xứng được chọn cùng độ dài mệnh đề và chiến lược phân nhánh.

Biểu diễn thời gian và đóng gói. SAT theo mốc thời gian phù hợp với khung thời gian vừa phải và miền bắt đầu đã được lọc. Tọa độ thứ tự nén tốt các so sánh trên miền nguyên. DDD mở rộng lựa chọn cho khung thời gian lớn khi có tập mốc đầy đủ hữu hạn, cận dưới hợp lệ và bước tinh chỉnh tăng nghiêm ngặt.

Khoảng cấm và biên thứ tự. Hai literal biên nén hiệu quả một khoảng cấm trên miền có thứ tự và đặc tả thứ tự hai chiều. Miền rời rạc dùng ánh xạ phần tử trước/sau; mỗi khoảng rời nhau bổ sung một cặp biên. Nhất quán cung tạo nhiều giá trị nhất trước SAT khi miền lớn hoặc đồ thị ràng buộc đủ dày.

12.4. Hướng nghiên cứu tiếp theo

1. **Lựa chọn bộ biểu diễn theo bài toán.** Xây bộ chọn có thể giải thích quyết định bằng các đặc trưng như n , k , độ chồng lấn, độ thừa của miền và số cận sẽ truy vấn; đánh giá bộ chọn so với bộ giải tốt nhất ảo.
2. **Đặc tả và kết hợp.** Phát triển thư viện mô-đun mà mỗi literal đầu ra có ngữ nghĩa máy đọc được, cho phép kiểm tự động việc ghép bộ đếm, liên kết biểu diễn, hàm mục tiêu và bộ giải mã.
3. **Chứng nhận toàn trình.** Nối ghi chứng minh của SAT/MaxSAT với bộ kiểm tra miền và chứng nhận cận dưới để giá trị tối ưu không phụ thuộc vào niềm tin ở bộ sinh.
4. **Trạng thái chia sẻ tổng quát.** Mô hình hóa việc chọn khối, cây ghép và ngưỡng cần hiện thực hóa (reification) như một bài toán tối ưu riêng; tìm cận dưới cho số trạng thái/bộ nối dưới các yêu cầu lan truyền khác nhau.
5. **Phương pháp kết hợp có phân công rõ.** Dùng CP/MIP để lọc miền hoặc tạo cận, SAT/MaxSAT để xử lý phép tuyển, nhưng đánh giá bằng thí nghiệm phân tích thành phần (ablation) để biết lợi ích đến từ mô-đun nào.
6. **Gói tái lập bền vững.** Dùng vùng chứa phần mềm, bản kê dữ liệu, mã băm, giấy phép và quy trình tái lập; tách kết luận đã công bố khỏi kết quả thăm dò.

12.5. Kết luận cuối

Giá trị của phép biểu diễn SAT nằm ở khả năng chọn đúng trạng thái trung gian để CDCL nhìn thấy cấu trúc của bài toán. Shared Counter, thanh ghi tam giác, biên thứ tự, biến quan hệ và rời rạc hóa động là các biểu hiện khác nhau của cùng một nguyên tắc: chỉ hiện thực hóa (reification) thông tin vừa đủ, đặt đặc tả rõ cho thông tin đó, rồi kiểm bằng cả chứng minh lẫn gói tái lập. Ba bước này chuyển một nghiên cứu tình huống thành một phương pháp có thể kiểm chứng, so sánh và dùng lại.

A Thư viện PySAT và các Bộ giải Chuyên ngành

Phụ lục này giới thiệu tổng quan về thư viện **PySAT** và các lớp bộ giải (**SAT**, **MaxSAT**, **Constraint Programming – CP**, và **Integer Linear Programming – ILP/MIP**) được sử dụng phổ biến trong nghiên cứu thiết kế phép biểu diễn và tối ưu hóa tổ hợp.

A.1. Tổng quan Công cụ PySAT (Python Toolkit for SAT Oracles)

A.1.1. Giới thiệu và Kiến trúc

Thư viện **PySAT** [43] là bộ công cụ mã nguồn mở dựa trên Python cho phép xây dựng mô hình, biểu diễn các ràng buộc logic phức tạp và tương tác trực tiếp với các SAT solver hàng đầu qua giao diện C-API. Thay vì phải làm việc với các tệp tin lưu đĩa dạng CNF (DIMACS format), PySAT liên kết trực tiếp vào bộ nhớ (in-memory bindings) giúp giảm thiểu chi phí I/O và hỗ trợ các luồng giải tăng dần (incremental solving).

Cấu trúc chính của PySAT bao gồm các mô-đun cốt lõi:

- `pysat.formula`: Quản lý công thức logic (CNF, WCNF).
- `pysat.card`: Xây dựng tự động các bộ biểu diễn ràng buộc đếm (Cardinality constraints).
- `pysat.pb`: Xây dựng bộ biểu diễn bất đẳng thức tuyến tính Boolean (Pseudo-Boolean constraints).
- `pysat.solvers`: Giao diện gọi trực tiếp các SAT solver (MiniSat, Glucose, CaDiCaL, Lingeling...).
- `pysat.examples`: Tích hợp các thuật toán tối ưu nâng cao như bộ giải MaxSAT RC2.

A.1.2. Minh họa Sử dụng PySAT

Ví dụ sau đây minh họa quy trình tạo công thức CNF, tích hợp ràng buộc đếm AtMostK thông qua Sorting Network (mạng sắp xếp) hoặc bộ đếm chuỗi (Sequential Counter), và gọi bộ giải GLUCOSE4:

```
from pysat.formula import CNF
from pysat.card import CardEnc, EncType
from pysat.solvers import Solver

# 1. Initialize CNF formula
cnf = CNF()
cnf.append([1, 2, 3]) # Clause: x1 v x2 v x3

# 2. Add Cardinality constraint: AtMost1(x1, x2, x3, x4)
# Use Sequential Counter (seqcounter) encoding
card_clauses = CardEnc.atmost(
    lits=[1, 2, 3, 4],
    bound=1,
    top_id=cnf.nv,
```

```

        encoding=EncType.seqcounter
    )
    cnf.extend(card_clauses)

# 3. Call SAT Solver (Glucose 4)
with Solver(name='g4', bootstrap_with=cnf) as solver:
    if solver.solve():
        print("SAT - Model found:", solver.get_model())
    else:
        print("UNSAT")

```

A.2. Các Bộ giải SAT và MaxSAT

A.2.1. Các Bộ giải SAT dựa trên CDCL

Các SAT solver hiện đại dựa trên kiến trúc **Conflict-Driven Clause Learning (CDCL)** [58] là động cơ giải quyết cốt lõi cho các mô hình CNF:

- **MiniSat** [32]: Bộ giải nền tảng có kiến trúc tối giản, sạch sẽ, làm chuẩn mực cho hầu hết các phát triển SAT solver sau này.
- **Glucose** [6]: Nổi tiếng với chỉ số *Literal Block Distance (LBD)* giúp đánh giá chính xác chất lượng mệnh đề xung đột và thực hiện dọn dẹp bộ nhớ hiệu quả.
- **CaDiCaL** [13]: Bộ giải CDCL hiện đại do Armin Biere phát triển, tích hợp sâu các kỹ thuật tiền xử lý (preprocessing) và trong khi giải (*in-processing*, *vivification*, *subsumption*) cùng cơ chế xuất chứng minh UNSAT (*proof logging* DRAT/LRAT).
- **Kissat** [13]: Bộ giải CDCL viết bằng C thuần túy, được tối ưu hóa ở mức mã máy để khai thác tối đa bộ nhớ đệm CPU (cache efficiency) và tốc độ lan truyền đơn vị (*unit propagation*).
- **CryptoMiniSat** [75]: Tích hợp thuật toán khử Gaussian (Gaussian Elimination) cho các hệ phương trình XOR, thích hợp cho các bài toán mật mã và đại số Boolean.

A.2.2. Các Bộ giải MaxSAT

Khi bài toán chứa cả các ràng buộc cứng (bắt buộc thỏa) và ràng buộc mềm (tối ưu hóa trọng số), các bộ giải MaxSAT được áp dụng:

- **RC2** [43]: Bộ giải MaxSAT dựa trên nói lỏng lõi xung đột (*core-guided MaxSAT solver*) được cài đặt trực tiếp trong PySAT.
- **Open-WBO** [60]: Thư viện MaxSAT mã nguồn mở mô-đun hóa cao, hỗ trợ nhiều thuật toán nói lỏng lõi và tìm kiếm dựa trên cận.

A.3. Các Bộ giải Quy hoạch Ràng buộc (CP Solvers)

Quy hoạch ràng buộc (**Constraint Programming – CP**) mở rộng khả năng biểu diễn sang các miền giá trị hữu hạn nguyên và các ràng buộc toàn cục (*global constraints*):

- **Google OR-Tools CP-SAT**: Bộ giải CP thể hệ mới kết hợp kỹ thuật sinh mệnh đề lười (*Lazy Clause Generation – LCG*) với động cơ CDCL SAT solver bên dưới. CP-SAT đạt hiệu năng vượt trội trong các bài toán lập lịch (*scheduling*) và đóng gói (*packing*).

- **Choco Solver:** Thư viện CP mã nguồn mở dựa trên Java, hỗ trợ phong phú các ràng buộc toàn cục (ALLDIFFERENT, CUMULATIVE, ELEMENT) và các chiến lược tìm kiếm tùy biến.
- **Gecode:** Hệ thống CP mã nguồn mở viết bằng C++ với hiệu năng cao, dựa trên kiến trúc nhân bản trạng thái (*cloning-based search*).

A.4. Các Bộ giải Quy hoạch Tuyến tính Số nguyên (ILP/MIP Solvers)

Phương pháp Quy hoạch tuyến tính số nguyên (**Mixed Integer Programming – MIP / ILP**) tiếp cận bài toán thông qua các bất đẳng thức tuyến tính và hình học lồi:

- **Gurobi Optimizer:** Bộ giải thương mại hàng đầu thế giới về tốc độ cho các bài toán MIP, QP và QCP.
- **SCIP (Solving Constraint Integer Programs)** [12]: Bộ giải mã nguồn mở mạnh mẽ nhất cho MIP và CIP, hỗ trợ mở rộng các quy tắc cắt (*cutting planes*) và phân nhánh (*branching rules*).
- **HiGHS / COIN-OR CBC:** Các bộ giải MIP mã nguồn mở phổ biến, thích hợp cho tích hợp vào các hệ thống thương mại và nghiên cứu rộng rãi.

A.5. Bảng Tổng hợp và Định hướng Lựa chọn Bộ giải

Bảng A.1 tóm tắt đặc tính kỹ thuật và phạm vi ứng dụng đề xuất cho từng lớp bộ giải.

Bảng A.1.: So sánh đặc tính kỹ thuật của các lớp bộ giải tổ hợp.

Lớp bộ giải	Dạng biểu diễn chính	Cơ chế xử lý chính	Bài toán phù hợp
SAT Solvers (CaDiCaL, Kissat)	Công thức CNF Boolean	CDCL, Unit Propagation, Conflict Learning	Mô hình logic thuần, ràng buộc phụ thuộc mạnh
MaxSAT Solvers (RC2, Open-WBO)	WCNF (Cứng + Mềm)	Core-guided Relaxation, SAT Oracles	Tối ưu hóa tổ hợp với mục tiêu tuyến tính
CP Solvers (OR-Tools CP-SAT)	Biến nguyên & Ràng buộc toàn cục	Lazy Clause Generation, Domain Filtering	Lập lịch sản xuất, đóng gói 2D/3D, phân bổ tài nguyên
ILP/MIP Solvers (Gurobi, SCIP)	Bất đẳng thức tuyến tính số nguyên	Branch-and-Bound, Cutting Planes, Simplex	Bài toán có ma trận thưa, mục tiêu số thực/nguyên rộng

Thuật ngữ Anh–Việt

Bảng T.1.: Thuật ngữ Anh–Việt.

Thuật ngữ tiếng Anh	Tương đương tiếng Việt và giải thích
ablation	thí nghiệm phân tích thành phần (ablation study): chỉ thay một thành phần để đo riêng tác động
arc consistency	nhất quán cung: mỗi giá trị còn ít nhất một giá trị hỗ trợ trên mỗi cung ràng buộc
assumption	giả định: literal tạm thời cho một lần gọi bộ giải
at-least-k	ràng buộc có ít nhất k literal đúng
at-most-k	ràng buộc có nhiều nhất k literal đúng
auxiliary variable	biến phụ có đặc tả ngữ nghĩa
backjump	nhảy lùi: quay lại một mức quyết định trước đó do phân tích xung đột
benchmark	bộ kiểm chuẩn: tập bài toán thử và quy trình đánh giá
binary decision diagram	biểu đồ quyết định nhị phân, viết tắt BDD
bound tightening	siết cận miền hoặc cận mục tiêu
branch-and-bound	tìm kiếm phân nhánh và cận
cactus plot	biểu đồ Cactus (Cactus plot): số bài giải được theo thời gian tích lũy
cardinality constraint	ràng buộc đếm số literal đúng
certified optimum	nghiệm tối ưu có cận dưới tương ứng
channeling	liên kết biểu diễn: các mệnh đề nối hai cách biểu diễn cùng một quyết định
clause learning	học mệnh đề từ xung đột
conflict-driven learning	clause học mệnh đề theo xung đột, viết tắt CDCL
connector	bộ nối: các mệnh đề ghép trạng thái do hai mô-đun cung cấp
consistency	mức nhất quán hoặc lọc miền mà phép biểu diễn/bộ lan truyền đạt được
constraint programming	quy hoạch ràng buộc, viết tắt CP
contract	đặc tả ngữ nghĩa mà một mô-đun cung cấp cho mô-đun sử dụng
cross-validation	kiểm định chéo: luân phiên tập tinh chỉnh và tập đánh giá
decoder	bộ giải mã mô hình Boolean thành nghiệm của bài toán gốc
direct encoding	Direct Encoding: một literal giá trị cho mỗi giá trị miền
dominance rule	quy tắc trội: loại nghiệm bị trội mà vẫn giữ ít nhất một nghiệm tối ưu
encoding	phép biểu diễn một ràng buộc hoặc mô hình thành CNF

Thuật ngữ tiếng Anh	Tương đương tiếng Việt và giải thích
equisatisfiable	đồng khả thỏa: hai công thức cùng có nghiệm hoặc cùng vô nghiệm
exactly-k	ràng buộc có đúng k literal đúng
external validity	hiệu lực khái quát: kết luận còn giữ ngoài tập thử hay không
feasibility	tính tồn tại nghiệm thỏa mọi ràng buộc cứng
feasible solution	nghiệm khả thi
fixpoint	điểm bất động: trạng thái mà áp dụng tiếp phép lan truyền không làm thay đổi miền hay phép gán
forbidden interval	khoảng cấm do một lựa chọn giá trị tạo ra
global constraint	ràng buộc toàn cục có ngữ nghĩa và bộ lan truyền riêng
incremental SAT	SAT tăng dần: dừng lại trạng thái bộ giải qua nhiều lần gọi
internal validity	hiệu lực nội tại: hiệu ứng quan sát đúng là do yếu tố đang xét
lazy clause generation	sinh mệnh đề lười: kết hợp lan truyền miền với lời giải thích dạng mệnh đề
literal	biến Boolean hoặc phủ định của biến đó
lower bound	cận dưới hợp lệ của giá trị mục tiêu
model checker	bộ kiểm tra trực tiếp nghiệm đã giải mã
non-preemption	tác vụ chạy trong một khoảng liên tục, không bị ngắt
order encoding	Order Encoding: chuỗi literal ngưỡng Order Encoding trong miền
performance profile	đồ thị tỷ lệ hiệu năng chuẩn hóa theo cấu hình tốt nhất
population	quần thể bài toán hoặc cấu hình mà kết luận hướng đến
proof logging	ghi chứng minh: xuất bằng chứng mệnh đề để bộ kiểm tra xác minh UNSAT
propagation complete	đầy đủ lan truyền trong phạm vi đã định nghĩa
pseudo-Boolean constraint	bất đẳng thức tuyến tính trên biến Boolean
random seed	hạt giống ngẫu nhiên: giá trị khởi tạo bộ sinh số giả ngẫu nhiên để tái lập một lần chạy
register	thanh ghi ngưỡng: biến phụ lưu trạng thái đếm ở một tiền tố hoặc một mô-đun
reification	hiện thực hóa (reification) chân trị: gán một literal với tính đúng/sai của một ràng buộc
relaxation	mô hình nối lỏng dùng để tạo cận hoặc phân tích
restart	khởi động lại nhánh tìm kiếm nhưng giữ các mệnh đề đã học
right-censored	dữ liệu bị giới hạn thời gian: chỉ biết thời gian chạy lớn hơn thời hạn
selection network	mạng lựa chọn: mạng bộ so sánh chỉ tạo các đầu ra cần thiết
shifted geometric mean	trung bình nhân dịch chuyển để tổng hợp thời gian chạy
sorting network	Sorting Network xây từ các bộ so sánh Boolean
soundness	tính đúng: mọi mô hình Boolean giải mã thành nghiệm hợp lệ
support encoding	Support Encoding: liệt kê các giá trị hỗ trợ cho một lựa chọn

Thuật ngữ tiếng Anh	Tương đương tiếng Việt và giải thích
symmetry breaking	phá vỡ đối xứng: giữ đại diện chuẩn tắc của mỗi quỹ đạo
symmetry breaking predicate	mệnh đề giữ ít nhất một đại diện của mỗi quỹ đạo
timeout	quá thời hạn: lần chạy vượt giới hạn thời gian
totalizer	Totalizer dạng đơn phân cho ràng buộc đếm/PB
trail	vết gán: dãy literal được gán cùng mức quyết định và lý do
unit propagation	lan truyền đơn vị
upper bound	cận trên đến từ một nghiệm khả thi
virtual best solver	bộ giải tốt nhất ảo: phép chọn hậu nghiệm cấu hình nhanh nhất cho từng bài toán thử
witness solution	nghiệm làm chứng: nghiệm cụ thể xác nhận một cận khả thi
witness variable	biến làm chứng: literal chỉ ra nhánh được chọn của một phép tuyển

Tài liệu tham khảo

- [1] K. I. Aardal, S. P. M. van Hoesel, A. M. C. A. Koster, C. Mannino, và A. Sassano. “Models and Solution Techniques for Frequency Assignment Problems”. Trong: *Annals of Operations Research* 153 (2007), trang 79–129. DOI: 10.1007/s10479-007-0178-0.
- [2] I. Abío, R. Nieuwenhuis, A. Oliveras, E. Rodríguez-Carbonell, và V. Mayer-Eichberger. “A New Look at BDDs for Pseudo-Boolean Constraints”. Trong: *Journal of Artificial Intelligence Research* 45 (2012), trang 443–480.
- [3] F. A. Aloul, I. L. Markov, và K. A. Sakallah. “Shatter: Efficient Symmetry-Breaking for Boolean Satisfiability”. Trong: *Proceedings of the 40th Design Automation Conference*. 2003, trang 836–839. DOI: 10.1145/775832.776042.
- [4] C. Ansótegui, M. L. Bonet, và J. Levy. “SAT-Based MaxSAT Algorithms”. Trong: *Artificial Intelligence* 196 (2013), trang 77–105.
- [5] R. Asín, R. Nieuwenhuis, A. Oliveras, và E. Rodríguez-Carbonell. “Cardinality Networks: A Theoretical and Empirical Study”. Trong: *Constraints* 16 (2011), trang 195–221.
- [6] G. Audemard và L. Simon. “Predicting Learnt Clauses Quality in Modern SAT Solvers”. Trong: *Proceedings of IJCAI*. 2009, trang 399–404.
- [7] F. Bacchus. “GAC via Unit Propagation”. Trong: *Principles and Practice of Constraint Programming – CP 2007*. 2007, trang 133–147.
- [8] O. Bailleux. “On the CNF Encoding of Cardinality Constraints and Beyond”. Trong: *arXiv preprint arXiv:1012.3853* (2010).
- [9] O. Bailleux và Y. Bouffhad. “Efficient CNF Encoding of Boolean Cardinality Constraints”. Trong: *Principles and Practice of Constraint Programming – CP 2003*. 2003, trang 108–122.
- [10] K. E. Batchier. “Sorting Networks and Their Applications”. Trong: *AFIPS Spring Joint Computer Conference*. 1968.
- [11] C. Bessiere, G. Katsirelos, N. Narodytska, C.-G. Quimper, và T. Walsh. “Decompositions of All Different, Global Cardinality and Related Constraints”. Trong: *Proceedings of the 21st International Joint Conference on Artificial Intelligence*. 2009, trang 419–424.
- [12] K. Bestuzheva và cộng sự. “The SCIP Optimization Suite 8.0”. Trong: *ACM Transactions on Mathematical Software* 49.2 (2023), trang 1–33.
- [13] A. Biere, K. Fazekas, M. Fleury, và M. Heisinger. “CaDiCaL, Kissat, Paracooba Entering the SAT Competition 2020”. Trong: *Proceedings of SAT Competition 2020 – Solver and Benchmark Descriptions*. 2020, trang 51–53.
- [14] A. Biere, M. Heule, H. van Maaren, và T. Walsh, eds. *Handbook of Satisfiability*. IOS Press, 2009.
- [15] M. Boffill, J. Coll, M. Garcia, J. Giráldez-Cru, G. Pesant, J. Suy, và M. Villaret. “Constraint solving approaches to the business-to-business meeting scheduling problem”. Trong: *Journal of Artificial Intelligence Research* 74 (2022), trang 263–301.
- [16] M. Boffill, J. Coll, P. Nightingale, J. Suy, F. Ulrich-Oltean, và M. Villaret. “SAT Encodings for Pseudo-Boolean Constraints Together with At-Most-One Constraints”. Trong: *Artificial Intelligence* 302 (2022), trang 103604. DOI: 10.1016/j.artint.2021.103604.
- [17] M. Boffill, J. Coll, J. Suy, và M. Villaret. “SMT Encodings for Resource-Constrained Project Scheduling Problems”. Trong: *Computers & Industrial Engineering* 149 (2020), trang 106777. DOI: 10.1016/j.cie.2020.106777.
- [18] N. Boland, M. Hewitt, L. Marshall, và M. Savelsbergh. “The Continuous-Time Service Network Design Problem”. Trong: *Operations Research* 65.5 (2017), trang 1303–1321. DOI: 10.1287/opre.2017.1624.
- [19] S. Brand, N. Narodytska, C.-G. Quimper, P. J. Stuckey, và T. Walsh. “Encodings of the SEQUENCE Constraint”. Trong: *Principles and Practice of Constraint Programming – CP 2007*. 2007, trang 210–224.
- [20] R. E. Bryant. “Graph-Based Algorithms for Boolean Function Manipulation”. Trong: *IEEE Transactions on Computers* C-35.8 (1986), trang 677–691.
- [21] J. Chen. “A New SAT Encoding of the At-Most-One Constraint”. Trong: *Workshop on Constraint Modelling and Reformulation*. 2010.
- [22] M. Chiarandini, L. Di Gaspero, S. Gualandi, và A. Schaerf. “The balanced academic curriculum problem revisited”. Trong: *Journal of Heuristics* 18.1 (2012), trang 119–148.
- [23] F. Clautiaux, J. Carlier, và A. Moukrim. “A New Exact Method for the Two-Dimensional Orthogonal Packing Problem”. Trong: *European Journal of Operational Research* 183.3 (2007), trang 1196–1211. DOI: 10.1016/j.ejor.2005.12.048.
- [24] F. Clautiaux, A. Jouglet, J. Carlier, và A. Moukrim. “A New Constraint Programming Approach for the Orthogonal Packing Problem”. Trong: *Computers & Operations Research* 35.3 (2008), trang 944–959. DOI: 10.1016/j.cor.2006.05.012.
- [25] S. A. Cook. “The Complexity of Theorem-Proving Procedures”. Trong: *Proceedings of the Third Annual ACM Symposium on Theory of Computing*. 1971, trang 151–158.
- [26] J. M. Crawford và A. B. Baker. “Experimental Results on the Application of Satisfiability Algorithms to Scheduling Problems”. Trong: *Proceedings of the Twelfth National Conference on Artificial Intelligence*. 1994, trang 1092–1097.

- [27] J. M. Crawford, M. L. Ginsberg, E. M. Luks, và A. Roy. “Symmetry-Breaking Predicates for Search Problems”. Trong: *Proceedings of the Fifth International Conference on Principles of Knowledge Representation and Reasoning*. 1996, trang 148–159.
- [28] N. X. Dao, P. A. Dang, và K. V. To. “A SAT-Based Exact Approach for the Minimum Order Frequency Assignment Problem”. Internal manuscript. 2026.
- [29] J. Davies và F. Bacchus. “Solving MaxSAT by Solving a Sequence of Simpler SAT Instances”. Trong: *Principles and Practice of Constraint Programming – CP 2011*. 2011, trang 225–239.
- [30] E. Demirović, N. Musliu, và F. Winter. “Modeling and Solving Staff Scheduling with Partial Weighted MaxSAT”. Trong: *Annals of Operations Research* 275 (2019), trang 79–99. doi: 10.1007/s10479-017-2693-y.
- [31] E. D. Dolan và J. J. Moré. “Benchmarking Optimization Software with Performance Profiles”. Trong: *Mathematical Programming* 91.2 (2002), trang 201–213.
- [32] N. Eén và N. Sörensson. “An Extensible SAT-Solver”. Trong: *Theory and Applications of Satisfiability Testing*. 2003, trang 502–518.
- [33] N. Eén và N. Sörensson. “Temporal Induction by Incremental SAT Solving”. Trong: *Electronic Notes in Theoretical Computer Science* 89.4 (2003).
- [34] N. Eén và N. Sörensson. “Translating Pseudo-Boolean Constraints into SAT”. Trong: *Journal on Satisfiability, Boolean Modeling and Computation* 2 (2006), trang 1–26.
- [35] K. Fazekas, M. Sinnl, A. Biere, và S. Parragh. “Duplex Encoding of Staircase At-Most-One Constraints for the Antibandwidth Problem”. Trong: *Integration of Constraint Programming, Artificial Intelligence, and Operations Research*. 2020, trang 186–204.
- [36] J. A. Gallian. “A Dynamic Survey of Graph Labeling”. Trong: *The Electronic Journal of Combinatorics, Dynamic Surveys* DS6 (2024). Twenty-seventh edition. doi: 10.37236/27.
- [37] I. P. Gent. “Arc Consistency in SAT”. Trong: *Proceedings of the 15th European Conference on Artificial Intelligence*. 2002, trang 121–125.
- [38] I. P. Gent, K. E. Petrie, và J.-F. Puget. “Symmetry in Constraint Programming”. Trong: *Handbook of Constraint Programming*. 2006, trang 329–376.
- [39] X. Guan, S. Zhang, R. Li, L. Chen, và W. Yang. “Anti-k-labeling of graphs”. Trong: *Applied Mathematics and Computation* 363 (2019), trang 124549.
- [40] A. Haberlandt, H. Green, và M. J. H. Heule. “Effective Auxiliary Variables via Structured Reencoding”. Trong: *arXiv preprint arXiv:2307.01904* (2023).
- [41] D. P. N. Hai, N. X. Dao, và K. V. To. “An exact approach solving No-hole Anti-k-labeling of graphs”. Trong: *Vietnam Journal of Computer Science* (2026).
- [42] E. Huang và R. E. Korf. “Optimal Rectangle Packing: An Absolute Placement Approach”. Trong: *Journal of Artificial Intelligence Research* 46 (2013), trang 47–87. doi: 10.1613/jair.3735.
- [43] A. Ignatiev, A. Morgado, và J. Marques-Silva. “PySAT: A Python Toolkit for Prototyping with SAT Oracles”. Trong: *Theory and Applications of Satisfiability Testing – SAT 2018*. 2018, trang 428–437.
- [44] A. Ignatiev, A. Morgado, và J. Marques-Silva. “RC2: An Efficient MaxSAT Solver”. Trong: *Journal on Satisfiability, Boolean Modeling and Computation* 11.1 (2019), trang 53–64. doi: 10.3233/SAT190116.
- [45] M. Iori, V. L. de Lima, S. Martello, F. K. Miyazawa, và M. Monaci. “Exact Solution Techniques for Two-Dimensional Cutting and Packing”. Trong: *European Journal of Operational Research* 289.2 (2021), trang 399–415. doi: 10.1016/j.ejor.2020.06.050.
- [46] S. Joshi, R. Martins, và V. Manquinho. “Generalized Totalizer Encoding for Pseudo-Boolean Constraints”. Trong: *Principles and Practice of Constraint Programming – CP 2015*. 2015, trang 200–209. doi: 10.1007/978-3-319-23219-5_15.
- [47] T. V. Kieu, C. L. Hoang, và K. V. To. “SAT-Based Approaches for Two-Dimensional Bin Packing: A Comprehensive Comparison with MIP and CP Methods”. Trong: *2025 17th International Conference on Knowledge and Systems Engineering*. 2025. doi: 10.1109/KSE68178.2025.11309458.
- [48] T. V. Kieu, C. L. Hoang, và K. V. To. “Solving the Two-Dimensional Single Stock Size Cutting Stock Problem with SAT and MaxSAT”. Trong: *arXiv preprint arXiv:2604.01732* (2026).
- [49] T. V. Kieu, D. Q. Le, và K. V. To. “Efficient SAT and MaxSAT Techniques for Solving the Two-Dimensional Strip Packing Problem”. Trong: *Pesquisa Operacional* 45 (2025). doi: 10.1590/0101-7438.2025.045.00297231.
- [50] T. V. Kieu, N. T. Nguyen, và K. V. To. “An Efficient SAT Encoding for Solving the Social Golfer Problem”. Trong: *RAIRO – Operations Research* 60.1 (2026), trang 173–199. doi: 10.1051/ro/2025155.
- [51] T. V. Kieu, P. C. Nguyen, B. G. Hoang, và K. V. To. “Compact SAT Encoding for Power Peak Minimization in Assembly Line Balancing”. Trong: *arXiv preprint arXiv:2512.11435* (2025).
- [52] T. V. Kieu, T. H. Nguyen, và K. V. To. “An Efficient MaxSAT-DDD Approach for Train Rescheduling via Precedence Propagation and Hybrid AMO Encodings”. Internal manuscript. 2026.
- [53] T. V. Kieu và K. V. To. “A Compact SAT Encoding for Non-Preemptive Task Scheduling on Multiple Identical Resources”. Trong: *Cybernetics and Information Technologies* 25.3 (2025), trang 104–122. doi: 10.2478/cait-2025-0025.
- [54] P. Kučera, P. Savický, và V. Vorel. “A Lower Bound on CNF Encodings of the At-Most-One Constraint”. Trong: *Theoretical Computer Science* 762 (2019), trang 51–73. doi: 10.1016/j.tcs.2018.09.003.

- [55] C.-M. Li, F. Manyà, và J. Planes. “MaxSAT, Hard and Soft Constraints”. Trong: *Handbook of Satisfiability*. IOS Press, 2009, trang 613–631.
- [56] M. Maher, N. Narodytska, C.-G. Quimper, và T. Walsh. “Flow-Based Propagators for the SEQUENCE and Related Global Constraints”. Trong: *Principles and Practice of Constraint Programming – CP 2008*. 2008, trang 159–174.
- [57] E. Malaguti và P. Toth. “A Survey on Vertex Coloring Problems”. Trong: *International Transactions in Operational Research* 17.1 (2010), trang 1–34. DOI: 10.1111/j.1475-3995.2009.00696.x.
- [58] J. Marques-Silva, I. Lynce, và S. Malik. “Conflict-Driven Clause Learning SAT Solvers”. Trong: *Handbook of Satisfiability*. IOS Press, 2009, trang 131–153.
- [59] R. Martins, S. Joshi, V. Manquinho, và I. Lynce. “Incremental Cardinality Constraints for MaxSAT”. Trong: *Principles and Practice of Constraint Programming – CP 2014*. 2014, trang 531–548.
- [60] R. Martins, V. Manquinho, và I. Lynce. “Open-WBO: A Modular MaxSAT Solver”. Trong: *Theory and Applications of Satisfiability Testing – SAT 2014*. 2014, trang 438–445.
- [61] J.-N. Monette, P. Flener, và J. Pearson. “A Propagator Design Framework for Constraints over Sequences”. Trong: *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*. 2014. DOI: 10.1609/aaai.v28i1.9113.
- [62] D. T. K. Nguyen, T. V. Kieu, và K. V. To. “SAT Encodings for Bandwidth Coloring: A Systematic Design Study”. Trong: *arXiv preprint arXiv:2602.08423* (2026).
- [63] D. T. K. Nguyen và K. V. To. “Proving Optimality for the Bandwidth Multicoloring Problem via SAT”. Trong: *arXiv preprint arXiv:2607.11121* (2026).
- [64] L. D. Nguyen, T. V. Kieu, và K. V. To. “Compact SAT and MaxSAT Encodings for Business-to-Business Meeting Scheduling with Idle-Time Balancing”. Trong: *arXiv preprint arXiv:2608.00614* (2026).
- [65] G. Pesant. “A Regular Language Membership Constraint for Finite Sequences of Variables”. Trong: *Principles and Practice of Constraint Programming – CP 2004*. 2004, trang 482–495.
- [66] G. Pesant, G. Rix, và L.-M. Rousseau. “A comparative study of MIP and CP formulations for the B2B scheduling optimization problem”. Trong: *International Conference on AI and OR Techniques in Constraint Programming*. 2015, trang 306–321.
- [67] J. Petke. *Bridging Constraint Satisfaction and Boolean Satisfiability*. Springer, 2015. DOI: 10.1007/978-3-319-21810-6.
- [68] T. Philipp và P. Reimer. “PBLib – A C++ Library for Pseudo-Boolean Optimization”. Trong: *Theory and Applications of Satisfiability Testing – SAT 2015*. 2015, trang 450–457.
- [69] J.-C. Régin. “A filtering algorithm for constraints of difference in CSPs”. Trong: *Proc. 12th National Conf. Artificial Intelligence (AAAI)*. 1994, trang 362–367.
- [70] A. Schutt, T. Feydy, P. J. Stuckey, và M. G. Wallace. “Solving RCPSP/max by Lazy Clause Generation”. Trong: *Journal of Scheduling* 16.3 (2013), trang 273–289. DOI: 10.1007/s10951-012-0285-x.
- [71] M. Sinnl. “A Note on Computational Approaches for the Antibandwidth Problem”. Trong: *Central European Journal of Operations Research* 29 (2021), trang 1073–1094. DOI: 10.1007/s10100-020-00688-4.
- [72] C. Sinz. “Towards an Optimal CNF Encoding of Boolean Cardinality Constraints”. Trong: *Principles and Practice of Constraint Programming – CP 2005*. 2005, trang 827–831.
- [73] T. Soh, K. Inoue, N. Tamura, M. Banbara, và H. Nabeshima. “A SAT-Based Method for Solving the Two-Dimensional Strip Packing Problem”. Trong: *Fundamenta Informaticae* 102.3–4 (2010), trang 467–487. DOI: 10.3233/FI-2010-314.
- [74] T. Soh, N. Tamura, và M. Banbara. “SAT-Based Method for the Two-Dimensional Bin Packing Problem with Rotation”. Trong: *2010 22nd IEEE International Conference on Tools with Artificial Intelligence*. 2010, trang 425–432.
- [75] M. Soos, K. Nohl, và C. Castelluccia. “Extending SAT Solvers to Cryptanalytic Problems”. Trong: *Theory and Applications of Satisfiability Testing – SAT 2009*. 2009, trang 244–257.
- [76] N. Tamura, A. Taga, S. Kitagawa, và M. Banbara. “Compiling Finite Linear CSP into SAT”. Trong: *Constraints* 14.2 (2009), trang 254–272.
- [77] H. V. Thanh, D. T. K. Nguyen, L. D. Do, T. H. Dao, T. T. Ninh, và K. V. To. “Exact k-Safe Labeling via Incremental SAT Solving”. Internal manuscript. 2026.
- [78] H. V. Thanh, D. D. Van, và K. V. To. “A SAT-Based Exact Approach for Radio k-Labeling”. Trong: *arXiv preprint arXiv:2607.19997* (2026).
- [79] M. Triska và N. Musliu. “An Improved SAT Formulation for the Social Golfer Problem”. Trong: *Annals of Operations Research* 194.1 (2012), trang 427–438. DOI: 10.1007/s10479-010-0702-5.
- [80] H. X. Truong, T. V. Kieu, và K. V. To. “Sequential Counter Encoding for Staircase At-Most-One Constraints”. Trong: *Proceedings of the 17th International Conference on Agents and Artificial Intelligence*. 2025, trang 164–175. DOI: 10.5220/0013124600003890.
- [81] H. X. Truong, T. V. Kieu, và K. V. To. “SAT Encoding for Set of At-Most-K Cardinality Constraints with Ladder Shape”. Revised manuscript submitted after major revision. 2026.
- [82] H. X. Truong, Q. H. Nguyen, N. X. Dao, T. V. Kieu, và K. V. To. “Solving At-Least Sequence Constraints in Nurse Rostering Problem Using SAT”. Trong: *2025 25th International Symposium on Communications and Information Technologies*. 2025, trang 397–402. DOI: 10.1109/ISCIT67082.2025.11231439.
- [83] H. X. Truong và K. V. To. “SAT Competition 2026 Benchmark Submission: Cyclic Anti-Bandwidth Feasibility Formulas”. Benchmark submission. 2026.

- [84] H. X. Truong và K. V. To. “Solving Cyclic Antibandwidth Problem by SAT”. Trong: *arXiv preprint arXiv:2601.04239* (2026).
- [85] G. S. Tseitin. “On the Complexity of Derivation in Propositional Calculus”. Trong: *Automation of Reasoning 2*. Original work published in 1968. Springer, 1983.
- [86] K. V. Tuyên. “Phương pháp biểu diễn tối ưu SAT cho các bài toán tối ưu hóa tổ hợp và lập lịch”. Luận văn thạc sĩ. Luận văn trong hồ sơ nhóm nghiên cứu, 2025.
- [87] F. Ulrich-Oltean, P. Nightingale, và J. A. Walker. “Selecting SAT Encodings for Pseudo-Boolean and Linear Integer Constraints”. Trong: *28th International Conference on Principles and Practice of Constraint Programming*. Tập 235. LIPIcs. 2022, 38:1–38:17. doi: 10.4230/LIPIcs.CP.2022.38.
- [88] T. Walsh. “General Symmetry Breaking Constraints”. Trong: *Principles and Practice of Constraint Programming – CP 2006*. 2006.
- [89] F. Wang và B. Wu. “Disproofs of two conjectures on no hole anti-n-labeling of graphs”. Trong: *Discrete Applied Mathematics* 300 (2021), trang 72–76.
- [90] N. Wetzler, M. J. H. Heule, và W. A. Hunt. “DRAT-trim: Efficient Checking and Trimming Using Expressive Clausal Proofs”. Trong: *Theory and Applications of Satisfiability Testing – SAT 2014*. 2014, trang 422–429.

Chỉ mục

- adaptive counter, 37
- anti-k-labeling, 74
- antibandwidth, 71
- arc consistency, 9
- arc-consistency preprocessing, 81
- at-most-one, 15

- B2B meeting scheduling, 56
- bandwidth coloring, 71
- bandwidth multicoloring, 71
- benchmark, 22
- bin packing, 63
- binary decision diagram, 15
- bảng chứng thực nghiệm, 89

- cactus plot, 22
- cardinality constraint, 15
- CDCL, 2
- channeling, 9, 43
- chứng nhận tối ưu, 89
- completeness, 9
- coordinate compression, 63
- cyclic antibandwidth, 71

- domain filtering, 56
- DPLL, 2
- Dynamic Discretization Discovery, 52

- encoding, 9
- exactly-k, 37

- forbidden interval, 71, 74, 81
- frequency assignment, 81

- giới hạn phương pháp, 89

- idle-time balancing, 56
- incremental SAT, 2

- k-safe labeling, 81

- Ladder AMK, 30
- lazy clause generation, 52
- lex-leader, 43

- MaxSAT, 2
- model checker, 2

- New Sequential Counter, 37
- No-hole Anti-k-labeling, 74
- non-overlap, 63
- non-preemption, 52
- Nurse Rostering, 52

- order encoding, 43

- PAR-2, 22
- performance profile, 22
- prefix variables, 74
- proof logging, 2
- pseudo-Boolean constraint, 15

- radio labeling, 81
- relation variable, 43
- rotation, 63

- SAT, 2
- scheduling, 52
- selection network, 89
- SEQUENCE constraint, 30
- sequential counter, 15
- Sequential Counter Ladder, 30
- shared counter, 30
- shifted geometric mean, 22
- sliding window, 30
- Social Golfer, 37
- sorting network, 15
- soundness, 9
- SparseSuffix, 56
- strip packing, 63
- symmetry breaking, 43

- timeout, 22
- totalizer, 15

- unit propagation, 9

- virtual best solver, 22