

Universidad de La Habana
Facultad de Matemática y Computación



Aplicación de la Lógica Computacional en la resolución de problemas

Autor:
Massiel Paz Otaño

Tutores:
Dr. Luciano García Garrido

Trabajo de Diploma
presentado en opción al título de
Licenciada en Ciencia de la Computación

Fecha

github.com/NinaSayers/Application-of-Computational-Logic-in-Problem-Solving.git

Dedicación

Agradecimientos

Agradecimientos

Opinión del tutor

Opiniones de los tutores

Resumen

Resumen en español

Abstract

Resumen en inglés

Índice general

| | |
|--|---|
| Introducción | 1 |
| 1. Estado del Arte | 4 |
| 2. Propuesta | 5 |
| 3. Detalles de Implementación y Experimentos | 6 |
| Conclusiones | 7 |
| Recomendaciones | 8 |

Índice de figuras

Ejemplos de código

Introducción

El desarrollo de la lógica computacional como disciplina se enmarca en la revolución tecnológica del siglo XX, impulsada por la necesidad de resolver problemas complejos en ámbitos como la inteligencia artificial, la verificación de hardware y software, y la optimización industrial. La creciente demanda de sistemas automatizados capaces de procesar restricciones y tomar decisiones eficientes llevó a la comunidad científica a explorar métodos formales para modelar y resolver problemas combinatorios. En este escenario, la teoría de la complejidad computacional emergió como un pilar fundamental, especialmente tras la identificación de la clase NP-Completo por Cook en 1971, que transformó la comprensión de los límites de la computación.

Los problemas con restricciones —aquellos que requieren satisfacer un conjunto de condiciones lógicas— han sido centrales en áreas como la planificación, la criptografía y el diseño de circuitos. El problema de satisfacibilidad booleana (SAT), demostrado por Cook como el primer problema NP-Completo, se convirtió en la piedra angular para estudiar la viabilidad de soluciones eficientes. Aunque los primeros algoritmos para SAT, como el método de Davis-Putnam (DP) y su evolución o Davis-Putnam-Logemann-Loveland (DPLL), sentaron las bases de los resolvers (solvers), su eficiencia se veía limitada por la explosión combinatoria en instancias complejas. La presencia de cláusulas unitarias, la selección subóptima de variables y el retroceso (backtrack) cronológico exponían claras debilidades, especialmente en problemas con miles de variables.

A pesar de los avances, los SAT resolvers (solvers) clásicos enfrentaban un desafío crítico: escalar sin sacrificar completitud. Esto motivó la búsqueda de mejoras heurísticas y estratégicas, como el aprendizaje de cláusulas y el retroceso (backtrack) no cronológico, que culminaron en el surgimiento del paradigma Conflict-Driven Clause Learning (CDCL). CDCL no solo optimizó la exploración del espacio de soluciones, sino que introdujo mecanismos para evitar repeticiones de conflictos, marcando un hito en la resolución práctica de problemas NP-Completo. Sin embargo, la eficacia de estos algoritmos depende en gran medida de estrategias de selección de variables, como VSIDS (Variable State Independent Decaying Sum) y DLIS (Dynamic Largest Individual Sum), cuyas ventajas comparativas siguen siendo objeto de debate.

Hoy, los SAT resolvers (solvers) basados en CDCL dominan aplicaciones críti-

cas, desde la verificación formal de chips hasta la síntesis de programas. No obstante, su rendimiento varía significativamente según el tipo de problema (p. ej., aleatorios vs. estructurados) y las heurísticas empleadas. Mientras VSIDS prioriza variables recientemente involucradas en conflictos —útil en problemas con alta estructura local—, DLIS enfatiza la frecuencia de aparición de literales, mostrando ventajas en dominios con distribución uniforme de restricciones. Esta dualidad plantea preguntas clave: ¿bajo qué métricas (tiempo de ejecución, memoria, escalabilidad) una estrategia supera a la otra? ¿Cómo influye la naturaleza del problema en su eficiencia?

Esta tesis aporta una comparación sistemática entre VSIDS y DLIS dentro de entornos CDCL, evaluando su desempeño en problemas heterogéneos (industriales, aleatorios y académicos). A diferencia de estudios previos, se integran métricas adaptativas que consideran no solo el tiempo de resolución, sino también el impacto de las cláusulas aprendidas y la distribución de conflictos. Además, se propone un marco teórico para clasificar problemas según su afinidad heurística, contribuyendo a la selección informada de algoritmos en aplicaciones reales.

Teóricamente, este trabajo profundiza en la relación entre estructura de problemas y heurísticas, enriqueciendo la comprensión de CDCL. Prácticamente, ofrece directrices para ingenieros y desarrolladores de resolvers (solvers), optimizando recursos en áreas como la verificación de software o la logística, donde minutos de mejora equivalen a ahorros millonarios.

Diseño teórico

- **Problema científico:** Ineficiencia de los SAT resolvers (solvers) ante problemas con distintas estructuras, asociada a la selección subóptima de variables.
- **Objeto de estudio:** Algoritmos CDCL con estrategias VSIDS y DLIS.
- **Objetivos:**
 - Analizar el impacto de VSIDS y DLIS en el rendimiento de CDCL.
 - Establecer correlaciones entre tipos de problemas y heurísticas eficaces.
- **Campo de acción:** Lógica computacional aplicada a la resolución de problemas con restricciones.
- **Hipótesis:** El rendimiento de VSIDS y DLIS varía significativamente según la densidad de restricciones, la presencia de patrones locales y el balance entre cláusulas aprendidas y originales.

El documento se organiza en cinco capítulos:

- **Fundamentos de SAT y NP-Complejidad:** Revisión teórica de problemas con restricciones y complejidad.

- **Evolución de los SAT resolvers (solvers):** Desde DP/DPLL hasta CDCL.
- **Heurísticas en CDCL:** VSIDS vs. DLIS, ventajas y limitaciones.
- **Metodología experimental:** Diseño de pruebas, métricas y casos de estudio.
- **Resultados y conclusiones:** Análisis comparativo y recomendaciones prácticas.

Esta investigación busca no solo esclarecer el debate entre VSIDS y DLIS, sino también sentar bases para el diseño de heurísticas adaptativas, impulsando la próxima generación de resolvers (solvers).

Capítulo 1

Estado del Arte

Capítulo 2

Propuesta

Capítulo 3

Detalles de Implementación y Experimentos

Conclusiones

Conclusiones

Recomendaciones

Recomendaciones