

Resumen

La programación con restricciones es un paradigma en el que las relaciones entre las variables se expresan en forma de restricciones. Es bien sabido que muchos de los problemas de la vida real se pueden modelar como Problemas de Satisfacción de Restricciones (CSPs). Debido a su amplia utilización en la resolución de estos problemas, se ha invertido mucho esfuerzo en incrementar la eficiencia de los algoritmos que resuelven CSPs. Sin embargo, muchas de estas técnicas asumen que el conjunto de variables, dominios y las restricciones involucrados en el CSP son completamente conocidos y fijos cuando el problema se modela. Esta es una fuerte limitación porque muchos problemas provienen de entornos inciertos y dinámicos, en los que el problema original, y en consecuencia su modelo CSP, pueden evolucionar debido al entorno, al usuario u otros agentes. En tales situaciones, una solución del problema original puede convertirse en inválida después de que se produzcan cambios en el problema original.

Existen dos estrategias principales para hacer frente a estas situaciones: reactivas y proactivas. Utilizar estrategias reactivas implica volver a resolver el CSP después de que una solución deje de ser una solución, lo cual conlleva un consumo de tiempo. Esta es una desventaja obvia, especialmente cuando tratamos con cambios a corto plazo. Esto también está motivado en [Verfaillie and Jussien, 2005], que es un importante estudio de la resolución de problemas con restricciones en entornos dinámicos e inciertos. En este trabajo, los autores afirman que un objetivo deseable en los marcos dinámicos e inciertos es:

Primera *“Limitar en la medida de lo posible la necesidad de sucesivas resoluciones online.”*

Debido a esta **primera** declaración, en esta tesis se desarrollan estrategias proactivas, que tratan de ofrecer una resistencia a las posibles modificaciones futuras del problema. Por lo tanto, estos métodos se aplican antes de que ocurran cambios en el problema original. Existen dos tipos principales de estrategias proactivas, las cuales pueden distinguirse en base a las características de las soluciones que se obtienen: *robustez* y *flexibilidad*. El concepto de flexibilidad implica modificaciones en la solución original, mientras que el concepto de robustez no. En [Verfaillie and Jussien, 2005], los autores hacen la **segunda** declaración como otro objetivo deseable.

Segunda *“Limitar en la medida de lo posible, los cambios en la solución producida.”*

Por esta razón, esta tesis se centra principalmente en la búsqueda de soluciones robustas, las cuales tienen una alta probabilidad de continuar siendo soluciones después de cambios en el CSP. Además, en [Verfaillie and Jussien, 2005] los autores

también mencionan como un interesante trabajo futuro la posibilidad de desarrollar estrategias proactivas que combinen las características de una solución de robustez y flexibilidad.

Tercera *“La producción de soluciones que sean a la vez robustas y flexibles, que tengan todas las posibilidades para resistir los cambios y puedan ser adaptadas fácilmente cuando no los resisten, es obviamente un objetivo deseable.”*

De acuerdo con este objetivo, expresado en la **tercera** declaración, en esta tesis se desarrollan estrategias que cumplen la condición de combinar la robustez y *estabilidad* (la estabilidad es un caso especial de flexibilidad). Por lo que nosotros conocemos, esta combinación todavía no se ha desarrollado para CSPs.

Las estrategias proactivas utilizan el conocimiento disponible acerca de los posibles cambios futuros con el fin de evitar o reducir al mínimo sus efectos. Sin embargo, para muchos de los problemas de la vida real la información sobre el entorno incierto y dinámico es desconocida o difícil de obtener. Para muchas estrategias proactivas que buscan soluciones robustas, la forma del algoritmo depende de un conocimiento detallado sobre el entorno dinámico. Como resultado, estos métodos no se pueden aplicar si la información requerida es desconocida. Por esta razón, el autor de [Hebrard, 2006] hace la **cuarta** declaración como una característica deseable de las estrategias que se enfrentan a dicho incierto y dinámico marco.

Cuarta *“Idealmente, no se requiere ningún conocimiento adicional sobre los datos utilizados para construir la clásica red de restricciones ni más experiencia para resolver el problema sin tener en cuenta la incertidumbre.”*

En esta tesis, tratamos de responder a una pregunta interesante en entornos dinámicos e inciertos: cuando no se conoce información adicional sobre los posibles cambios en el problema, es posible definir lo que es la robustez de una solución de un CSP y desarrollar algoritmos apropiados?. Nosotros creemos que es posible y justificable extraer algunas suposiciones limitadas (e intuitivamente razonables) sobre el dinamismo de los problemas para los que el orden sobre los elementos del dominio es importante. Por lo tanto, en este trabajo presentamos estrategias para hacer frente a tipos de problemas que son, por lo tanto, consistentes con la **cuarta** declaración. El cumplimiento de estas cuatro declaraciones ha sido la motivación y el objetivo principal del trabajo presentado en esta tesis.