

Inteligencia Artificial

Andrés Gómez de Silva Garza

Proyecto de Programación #3

El proyecto se puede (y se recomienda) hacer en equipos de hasta cuatro personas (los integrantes pueden ser diferentes de los que se juntaron para los proyectos de programación anteriores en el semestre). Es la responsabilidad de cada uno de los integrantes del equipo asegurarse de que la calidad de cada una de las componentes del proyecto sea suficiente para asegurar la obtención de una buena calificación. Poder completar el proyecto exitosamente requerirá de una buena organización, planeación, y coordinación entre todos los integrantes del equipo por anticipado, pues no es un proyecto trivial, así como del compromiso con el resto del equipo por parte de cada integrante. La correcta elaboración del proyecto también requerirá de la investigación independiente de algunos temas que no se hayan tratado en clase, o que se hayan tocado sólo brevemente.

La entrega consiste de dos partes:

- un programa escrito en SWI-Prolog (comentado/documentado adecuadamente para entender su funcionamiento y diseño), listo para ejecutarse y
- una presentación oral ante el resto del salón (de 30 minutos de duración), apoyándose en la proyección de una presentación de PowerPoint que incluya los puntos clave del proyecto a exponer, en la que se tiene que explicar el proyecto que se realizó (la teoría detrás de él), el diseño del programa (es decir, cómo se implementó/aplicó la teoría) y mostrar el programa en acción (es decir, correrlo en vivo durante la presentación para mostrar su funcionamiento ejemplificado a través de varios ejemplos de problemas de prueba); todos los integrantes del equipo deben participar en la presentación oral.

Se asignará una sola calificación que tome en cuenta las dos partes del proyecto. Se va a usar la rúbrica para presentaciones orales para calificar las dos partes. Se tiene que entregar (en un solo envío por equipo, a través del correo electrónico) tanto el programa en SWI-Prolog (recordando cambiarle su extensión a .txt para que no sea eliminado por el firewall del ITAM) como el archivo de PowerPoint a más tardar en la fecha señalada por su profesor.

Posibles proyectos (cada equipo debe escoger uno de ellos...no importa si varios equipos hacen el mismo, aunque cada equipo debe trabajar por separado, lo cual debe verse reflejado en un diseño diferente de cada versión):

1. Programar un mini-Router y realizar experimentos con él. Router es un sistema de inteligencia artificial que usa múltiples estrategias de razonamiento para planear rutas dentro del plantel de Georgia Tech. Su mini-Router debe hacer lo mismo pero en un dominio diferente (ejemplos posibles: la red del metro de la Ciudad de México o alguna otra ciudad con una red de metro de tamaño comparable, la red de carreteras federales entre las ciudades capitales y principales de México o un país de tamaño comparable, la red de conexiones aéreas en México o alguna otra región equivalente). Router combina aspectos de razonamiento basado en casos con razonamiento a través de un modelo/mapa organizado jerárquicamente y se le podría agregar búsqueda A* (cuando debe buscar dentro de una sola zona dentro de su mapa jerárquico). El tipo de experimentos interesantes que se podrían realizar con él se podrían diseñar para contestar preguntas como: ¿Alguno de los métodos de razonamiento que utiliza el sistema es más eficiente que los otros? ¿Alguno de

los métodos de razonamiento que utiliza el sistema garantiza la obtención de soluciones de mejor calidad que los otros? ¿Integrar los tres métodos de razonamiento es mejor que utilizar cualquiera de ellos por separado? ¿A partir de qué momento resulta ser contraproducente seguir almacenando casos específicos de problemas resueltos (y cómo medirlo para que un sistema sea capaz de auto-monitorearse y decidirlo)? Etc. Para realizar este proyecto parte de la investigación independiente que se tendrá que realizar consistirá en la obtención y captura del conocimiento geográfico relevante (por ejemplo, coordenadas, distancias, tiempos de recorrido, conexiones) que se necesite para describir el espacio dentro del cual se van a planear las rutas. Otra parte de la investigación independiente tiene que ver con elegir una función heurística adecuada (lo cual puede depender de la disponibilidad de datos adecuados) para la ejecución del algoritmo A*.

2. Diseñar e implementar un algoritmo evolutivo para asignar el calendario de partidos de una liga de fútbol (estilo europeo, sin considerar liguitas u otras rondas especiales que no sean el torneo de liga básico). La asignación ideal debe estar sujeta a una diversidad de restricciones, tanto duras (rígidas) como suaves: cada equipo se debe enfrentar a cada uno de los demás equipos exactamente dos veces en el torneo, una vez en casa y una vez de visitante; idealmente los dos enfrentamientos entre cada par de equipos no deben ser muy cercanos en el tiempo; se deben alternar partidos en casa y como visitante; cuando una ciudad tiene más de un equipo se debe tratar de asegurar que en cada jornada haya un partido en esa ciudad (o, en otras palabras, que si un equipo de la ciudad está jugando en casa en esa jornada, el/los otro(s) equipo(s) de la misma ciudad esté(n) jugando como visitante(s)); la distancia total que necesita viajar cada equipo se puede tratar de minimizar para ahorrar en gastos de combustible/boletos; etc.

3. Programar algún algoritmo heurístico genérico de propagación de restricciones para la resolución de problemas de criptoaritmética. Parte de la investigación independiente relacionada con este proyecto tiene que ver con proponer, programar y probar heurísticas que permitan resolver este tipo de problemas (criptoaritmética) de forma eficiente (garantizando que se llegue a la solución final, si la hay, pero no haciendo una búsqueda "de fuerza bruta", exhaustiva, sino dirigida). Otra parte de la investigación independiente que se puede requerir es hacer un análisis teórico (o llevar a cabo la programación y la prueba de diversas estrategias de búsqueda heurística o avanzada (por ejemplo, algoritmos evolutivos o recocido simulado)) para determinar qué algoritmo(s) de búsqueda es(son) más apropiado(s) para resolver este tipo de problemas.

4. Programar un mini-Autognotic (Autognotic, también conocido como meta-Router, es una versión de Router que tiene un modelo de sus propios métodos de razonamiento, por lo que puede, basándose en monitorear su propio razonamiento y los resultados obtenidos, sugerir auto-modificaciones a su forma de razonar). Su profesor les puede proporcionar algunos artículos descriptivos del meta-razonamiento que lleva a cabo Autognotic en los cuales pueden inspirarse durante las investigaciones independientes que tendrán que hacer para realizar esta versión del proyecto.

5. Adaptar el código que desarrollaron para el Proyecto de Programación 2 (jugar dominó usando búsqueda minimax con poda alfa-beta) para tener un sistema que ahora sea capaz de jugar alguna variante de pókar (bien) contra un contrincante potencialmente humano o potencialmente computacional. Parte de la investigación independiente que tendrán que realizar en esta versión del proyecto consiste en sustituir la función heurística del dominó por alguna parecida pero aplicable al pókar, y de su algoritmo de predicción de las fichas (en este caso, cartas) que tiene el oponente. Se permite además hacerle mejoras a su implementación de minimax con poda alfa-beta, en caso necesario, para elaborar esta versión del proyecto.

6. Algún otro proyecto equivalente en alcance, nivel de dificultad, cantidad de investigación independiente requerida, etc. (su propuesta tiene que ser aprobada por el profesor).

Recuerde que, para que el programa se considere “una solución de inteligencia artificial”, tiene que ser lo más genérico posible (aplicable a una variedad amplia de problemas relacionados y/o fácil de adaptar para lograr aplicarlo así) y estar diseñado basándose lo más posible en un análisis cuidadoso que lleve a una teoría formal, sistemática, algorítmica (en general computacional), claramente especificada y lo más completa posible acerca del dominio de aplicación del sistema.