

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. 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 tres partes:

- un programa escrito en SWI-Prolog (comentado/documentado adecuadamente para entender su funcionamiento y diseño), listo para ejecutarse,
- 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 (todos los integrantes del equipo deben participar en la presentación oral), y
- un reporte escrito documentando la elaboración del proyecto (estructurado como un artículo científico o una mini-tesis, incluyendo conclusiones y bibliografía) y lo que se aprendió de las investigaciones independientes que se hayan tenido que hacer para realizar el proyecto.

De estas tres partes, las primeras dos se calificarán juntas y la tercera se calificará por separado. La calificación final del proyecto consistirá del promedio de las dos calificaciones, dándole el mismo peso a cada una de ellas. Las dos rúbricas que se van a utilizar para calificar las últimas dos partes estarán disponibles a través de ComunidadITAM. La rúbrica para la presentación oral se va a usar para calificar tanto la presentación oral como el programa (pues algunos de los renglones de dicha rúbrica son específicos al material visual o la forma de exponer, pero otros se pueden interpretar de manera más amplia, por ejemplo los que tienen que ver con la cantidad de detalle y con la inclusión de suficientes ejemplos en la presentación). La calificación de la presentación oral podría ser diferente para cada integrante de un equipo. En los demás aspectos se calificará de manera pareja a todos los integrantes de cada equipo.

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, aplicado a algún dominio diferente del original, que era el plantel de Georgia Tech, y hacer experimentos con él. Router es un sistema de inteligencia artificial que usa múltiples estrategias para planear rutas. Combina aspectos de razonamiento basado en casos con razonamiento basado en modelos (en este caso un mapa organizado jerárquicamente) y con 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 resolver este proyecto se puede aprovechar tanto los datos que se hayan obtenido (por ejemplo, metros, calles, carreteras o conexiones aéreas de algún contexto mexicano) como el código que se haya producido para hacer el Proyecto de Programación #2.

2. Diseñar y programar un sistema multi-agente que sirva para simular la forma en la que los vehículos turísticos recorren los parques nacionales africanos para realizar un safari, y realizar experimentos con él para observar qué sucede bajo distintas suposiciones. Habrá que simular agentes de distintos tipos (por ejemplo, animales, vehículos, etc.) y utilizar variables con valores realistas para dotar a los agentes de distintas personalidades. Algunos experimentos que se podrían realizar deberían servir para contestar preguntas como las siguientes. ¿Se tiene un mayor impacto negativo sobre el ecosistema completo si se permite que los vehículos circulen a campo traviesa comparado con la circulación exclusiva en los caminos existentes? ¿Se forman más o menos agrupamientos de vehículos turísticos, en promedio, bajo alguna de las dos posibilidades de circulación antes mencionadas? ¿Cuál es la densidad óptima de vehículos turísticos para obtener beneficios (por ejemplo, ganancia económica y difusión de impresiones positivas acerca de los espectáculos naturales del país) sin causar demasiados daños (por ejemplo, destrucción permanente del medio ambiente e interferencia excesiva en la vida natural de los animales)? ¿Qué diferencias hay en los resultados obtenidos si suponemos una comunicación instantánea y de alcance ilimitado (por ejemplo, a través de transmisiones por radio), o instantánea pero dirigida (por ejemplo, a través de teléfonos celulares), o sólo visual, de la presencia de animales en distintos lugares dentro del parque? Etc. Todo esto se debe explorar bajo distintas suposiciones acerca de las características de los animales y vehículos y acerca de la densidad de los mismos (por ejemplo, en la temporada seca puede haber menos de los dos que en la temporada de lluvias o ciertas especies de animales pueden naturalmente congregarse en grupos grandes mientras que otras son completamente individualistas, algunas especies son más miedosas que otras cuando se aproxima un vehículo, pero hasta cierto punto pueden ser las que más buscan los turistas en los vehículos, algunos vehículos son más agresivos y están dispuestos a arriesgarse a ser multados si rompen las reglas de circulación para ver/buscar animales mientras que otros no, etc.), así como la interconectividad de los mismos (el comportamiento de algunos individuos de ciertas especies podría influir en el comportamiento de los individuos de otras especies, pero no necesariamente siempre, y probablemente depende de la distancia geográfica así como de las especies específicas).

3. 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 un par de equipo 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.

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 de Autognotic.

5. 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, claramente especificada acerca del dominio de aplicación del sistema.