



X encuentro
Participación de la
Mujer
en la Ciencia



GENERACION DE INSTANCIAS DE PRUEBAS PARA UN MODELO DE REDUCCION DE COLISIONES EN NAVEGACION CON RESTRICCIONES

Josué Jacob Oregel Picón, Héctor José Puga Soberanes, Juan Martín Carpio Valadez, Manuel Ornelas Rodríguez, Víctor Manuel Zamudio Rodríguez, Luis Ernesto Mancilla Espinoza

División de Estudios de Posgrado e Investigación, Instituto Tecnológico de León, León, Gto.,
{josue-oregel,hector.puga}@itleon.edu.mx

RESUMEN

Reducir la cantidad de colisiones entre objetos móviles es un problema de optimización bajo la consideración de que las colisiones generan un impacto negativo en sistemas de navegación reales. Para problemas de navegación se encuentran propuestas conocidas como VRP y AGV que buscan encontrar soluciones a problemas de enrutamiento y planificación de tareas en sistemas de navegación con diferentes características. En este trabajo se presenta un generador de instancias de pruebas para problemas de colisiones entre objetos móviles en sistemas de navegación a los que hemos llamado *Reducing Collisions in Constrained Navigation (RCCN)*. Estas instancias son útiles para analizar el rendimiento de técnicas de optimización inteligente aplicadas a problemas RCCN utilizando un modelo para detección de colisiones entre objetos móviles que sirve como plataforma para la reducción de las mismas.

1. INTRODUCCIÓN

En este trabajo se entiende un sistema de navegación como la composición de sistema que contiene un entorno de navegación, rutas para navegar, objetos navegantes (o móviles) y una planificación para navegar por las rutas.

Es deseable que en un sistema de navegación inteligente se tenga la capacidad de identificar las restricciones de un entorno de navegación y efectuar tareas de enrutamiento y planificación con autonomía. Para estos escenarios de problemas existen enfoques conocidos en el estado del arte, como la navegación de *Objetos Móviles Autónomos (Autonomous Mobile Objects, AMOs)* en problemas de satisfacción de restricciones como VRP y AGV [1][4].

Una clase específica de problemas de navegación son un conjunto de problemas de *Reducción de Conflictos de Navegación (RECONA)* [3], donde se consideran conflictos como congestiones, puntos muertos, colisiones con elementos estáticos, colisiones con elementos en movimiento. Como parte de los problemas RECONA ubicamos un subconjunto de problemas al que hemos llamado *Reducción de Colisiones en Navegación con Restricciones (Reducing Collisions in Constrained Navigation, RCCN)*, que considera en sus restricciones el tratamiento de eventos de colisión entre objetos móviles en sistemas de navegación.

Un modelo para detección de colisiones en sistemas de navegación con restricciones de problemas *RCCN* que facilita una plataforma para la reducción de las colisiones en el sistema mediante la implementación de técnicas de optimización inteligente utiliza dos conjuntos de datos como entrada:



X encuentro
Participación de la
Mujer
en la Ciencia



- Una “*Matriz de Conectividad Pesada*” (M_c) de tamaño $n \times n$, que es una matriz de adyacencia con pesos obtenida de un grafo que representa al entorno de navegación. Los pesos en las entradas de esta matriz representan la información de conectividad y tiempo de recorrido entre los n nodos del sistema de navegación.
- Un conjunto de m rutas del sistema de navegación.

Usando estos datos de entrada el modelo utiliza tres componentes para generar una *Matriz de Colisiones* (M_{col}) de tamaño m rutas por n nodos, que permite detectar el traslape en el tiempo de ocupación de nodos de rutas por parte de objetos móviles en el sistema de navegación, lo que se define como una colisión, y facilita una estrategia para la reducción de la cantidad de colisiones en el sistema mediante la configuración de tiempo de espera en nodos de las rutas de navegación. Las tres componentes que se usan para calcular las entradas de la matriz M_{col} son: tiempo de recorrido original (t_o), tiempo de espera en el nodo (t_e) y tiempo de retardo para la llegada al nodo (t_r).

2. GENERACIÓN DE INSTANCIAS DE PRUEBA

Una vez que se ha probado la utilidad del modelo en instancias de pruebas básicas [3], ha surgido la necesidad de probarlo con instancias de pruebas con una cantidad de eventos de colisión mayor, para poder realizar pruebas de rendimiento en la aplicación de técnicas de optimización inteligente, por ejemplo heurísticas basadas en *Particle Swarm Optimization (PSO)* y *Artificial Immune Systems (AIS)*.

En el estado del arte sobre navegación hay instancias de pruebas para problemas como VRP, AGV y algunas variantes de ambos. Sin embargo, para problemas RCCN es necesario contar con instancias de pruebas diseñadas para el problema de colisiones entre objetos móviles y este tipo de instancias no se han encontrado en el estado del arte, dando lugar a la necesidad de construir un generador de instancias para problemas RCCN propio. El generador de instancias de pruebas aquí presentado, entrega dos arreglos de datos por instancia que sirven como insumo para el modelo para problemas RCCN: una *Matriz de Conectividad Pesada* M_c y un conjunto de m rutas de navegación. El generador se explica en dos partes: generación de rutas de navegación y generación de tiempos de recorrido para las rutas del sistema de navegación.

2.1. GENERACIÓN DE RUTAS DE NAVEGACIÓN

El generador de instancias de pruebas se basa en una nube de n puntos (o nodos) para la generación de rutas de navegación. Las rutas de navegación están definidas como una sucesión de nodos y cada ruta se compone de una cantidad independiente de nodos, lo que permite la aplicación de una entre dos opciones de configuración: la misma cantidad de nodos o cantidades diferentes de nodos para las rutas.

Si se considera el conjunto de rutas generadas como un sistema de navegación, entonces el número de aristas totales A_t requeridas para generar el sistema se calcula a partir de la cantidad s_k de nodos de una ruta k (Ecuación 1):

$$A_t = \sum_{k=1}^m (s_k - 1) \quad (1)$$



X encuentro
Participación de la
Mujer
en la Ciencia



Por otro lado, la cantidad de aristas útiles A_u para generar rutas a partir de una nube de n puntos es la que se presenta en la Ecuación 2:

$$A_u = \frac{n(n-1)}{2} \quad (2)$$

Si $A_u \geq A_t$, teóricamente es posible con los n nodos del grafo completo generar el sistema de navegación con A_t aristas, de otra forma, el sistema no se podrá generar. El Algoritmo 1 presenta el pseudocódigo para la generación de rutas de navegación.

Algoritmo 1 Pseudo-código para la generación de rutas de navegación

Entrada: n : cantidad de nodos del sistema de navegación

Entrada: m : cantidad de rutas que se van a generar

Entrada: $s_{(R_{(k)})}$: cantidad de nodos de $R_{(k)} \mid 1 \leq k \leq m$ (el valor s es independiente para cada $R_{(k)}$)

```
1: para cada  $R_{(k)}$  desde 1 hasta  $m$  hacer
2:   seleccionar aleatoriamente y asignar un  $nodo_1$  (de inicio) a  $R_{(k)}$ 
3:   para cada  $i$  desde 2 hasta  $s_{(R_{(k)})}$  hacer
4:     repetir
5:       seleccionar aleatoriamente un  $nodo_i$ 
6:       si  $nodo_i$  es diferente de  $nodo_{i-1}$  (nodo anterior en la ruta) entonces
7:         si no existe una arista entre  $nodo_i$  y  $nodo_{i-1}$  entonces
8:           asignar  $nodo_i$  a  $R_{(k)}$ 
9:         fin si
10:      fin si
11:   hasta que ha sido asignado un  $nodo_i$  a  $R_{(k)}$ 
12:   fin para
13: fin para
14: devolver un conjunto de  $m$  trayectorias de ruta sobre el mapa de rutas usado.
```

2.2. GENERACIÓN DE TIEMPO DE RECORRIDO DE LAS RUTAS DE NAVEGACIÓN

Para generar colisiones usando las rutas generadas, se ubican los nodos comunes y se asigna tiempo de recorrido a cada segmento de las rutas, de tal forma que la suma de tiempo de recorrido hasta los nodos comunes sea la misma para cada ruta involucrada. El algoritmo 2 presenta el pseudocódigo para la asignación de tiempos de recorrido a los segmentos de ruta del sistema de navegación.



X encuentro
Participación de la
Mujer
en la Ciencia



Algoritmo 2: Pseudocódigo para la asignación de tiempo de recorrido de rutas

Entrada: m rutas de navegación

```
1: para cada  $R_k$  desde 1 hasta  $m$  hacer
2:   para cada  $n_{(k,i)}$  desde 1 hasta  $s_k$  hacer
3:     si  $n_{(k,i)}$  es un nodo común de  $R_k$  y otra ruta entonces
4:       aumentar el contador de cantidad de nodos comunes en  $R_k$ 
5:     fin si
6:   fin para
7: fin para
8:  $mnc$  = índice de la ruta con la mayor cantidad de nodos comunes
9: asignar un tiempo de recorrido igual a cero a  $n_{(mnc,1)}$ 
10: para cada  $n_{(mnc,i)}$  desde 2 hasta  $s_{mnc}$  hacer
11:   asignar aleatoriamente un tiempo de recorrido a  $n_{(mnc,i)}$ 
12: fin para
13: para cada  $R_k$  desde 1 hasta  $m$  excepto  $mnc$  hacer
14:   para cada  $n_{(k,i)}$  desde 1 hasta  $s_k$  hacer
15:     si  $n_{(k,i)}$  es un nodo común entre  $R_k$  y  $R_{mnc}$  entonces
16:       generar un tiempo de recorrido hasta  $n_{(k,i)}$  igual al de  $n_{(mnc,i)}$ 
17:     fin si
18:   fin para
19: fin para
20: si aún hay nodos comunes entonces
21:   para cada  $R_k$  desde 1 hasta  $m - 1$  excepto  $mnc$  hacer
22:     para cada  $R_l$  desde  $R_{k+1}$  hasta  $m$  excepto  $mnc$  hacer
23:       para cada  $n_{(k,i)}$  desde 1 hasta  $s_k$  hacer
24:         para cada  $n_{(l,i)}$  desde 1 hasta  $s_l$  hacer
25:           si  $n_{(k,i)} = n_{(l,i)}$  entonces
26:             generar un tiempo de recorrido hasta  $n_{(k,i)}$  igual al de  $n_{(l,i)}$ 
27:           fin si
28:         fin para
29:       fin para
30:     fin para
31:   fin para
32: fin si
33: si hay trayectos sin tiempos de recorrido entonces
34:   asignar tiempo de recorrido para los trayectos faltantes
35: fin si
36: devolver una Matriz de Conectividad Pesada  $Mc$ 
```

3. INSTANCIAS DE PRUEBAS GENERADAS

La generación de instancias de pruebas se llevó a cabo con un programa en Java [5] que entrega dos archivos para cada instancia: un archivo .mcc que contiene la Matriz de Conectividad Pesada Mc y un archivo .rtc que contiene m rutas de navegación. Estos dos archivos son las dos estructuras de datos que se usan como insumo en el modelo para problemas RCCN. Las instancias tienen la nomenclatura: **#N-#R-#C-ver#**, donde N hace referencia al número de nodos, R al número de rutas, C al número de colisiones del sistema y puesto que es posible generar diferentes mapas de rutas con la misma configuración de nodos, ver hace referencia a la versión de la instancia generada, comenzando con 1. El valor de semilla utilizado en el método Random del programa Java ha sido almacenado en los archivos generados para brindar la posibilidad de reproducir cada instancia de pruebas. Algunos ejemplos de nomenclatura de instancias de pruebas generadas se muestran en la Tabla 1



X encuentro
Participación de la
Mujer
en la Ciencia



Nombre de Instancia	Nodos por ruta generados	Semilla Java Random
100N60R-676C-ver1	Entre 60 y 90	1242326594992500141
150N-90R-941C-ver1	Entre 80 y 105	242298037008080666
200N-80R-1089C-ver1	Entre 120 y 200	-7332834085541032903
250N-200R-4879C-ver1	Entre 100 y 130	3618874879153313709
300N-120R-1739C-ver1	Entre 200 y 300	6519364232554299876
400N-300R-10541C-ver1	Entre 150 y 250	2333333709128786646
500N-300R-10293C-ver1	Entre 250 y 390	4932274200351035357
1000N-650R-75841C-ver1	Entre 700 y 770	-5907896972978185640
2000N-1000R-91390C-ver1	Entre 1,000 y 1,500	-9174493999035787771

Tabla 1. Ejemplos de nomenclatura de instancias de pruebas RCCN generadas con un programa en Java.

4. CONCLUSIONES

La presencia de conflictos, como colisiones, en sistemas de navegación se considera un problema debido al impacto negativo que los conflictos provocan en los sistemas reales. En sistemas de navegación donde la cantidad de colisiones entre móviles es grande puede ser necesario contar con herramientas de planificación que mejoren el desempeño del sistema.

En este trabajo se ha presentado un generador de instancias de pruebas con las características de problemas RCCN, que son la base para comparar el rendimiento de técnicas de optimización inteligente aplicadas a problemas RCCN. El generador construye instancias en dos pasos: generación en forma aleatoria de rutas en sistemas de navegación de n nodos y generación de colisiones en nodos comunes entre las rutas asignando tiempo de recorrido de las rutas.

Actualmente se está trabajando en heurísticas basadas en técnicas de optimización como PSO y AIS aplicadas a las instancias de prueba generadas para reducir la cantidad de colisiones en el sistema de navegación.

AGRADECIMIENTOS

Los autores agradecen el soporte del Consejo Nacional de Ciencia y Tecnología (CONACYT) y de la Dirección General de Educación Superior Tecnológica (DGEST), proyecto número 4572.12-P.

BIBLIOGRAFÍA

- [1] Gendreau, M.; Potvin, J.; Bräumlaysy, O.; Hasle, G. & Løkketangen, A. Metaheuristics for the vehicle routing problem and its extensions: A categorized bibliography. *The vehicle routing problem: Latest advances and new challenges*, Springer, **2008**, 143-169.
- [2] Oracle, Inc. Java Class Random. <http://docs.oracle.com/javase/1.4.2/docs/api/java/util/Random.html>, **2013**.
- [3] Oregel, Josué & Puga, Héctor. Propuesta de Técnicas de Inteligencia Artificial Para la Reducción de Colisiones en un Problema de Navegación. *IX Encuentro Participación de la Mujer en la Ciencia, Centro de Investigaciones en Óptica, A. C.*, **2012**, S4-ING34.
- [4] Qiu, L.; Hsu, W.; Huang, S. & Wang, H. Scheduling and routing algorithms for AGVs: a survey. *International Journal of Production Research*, Taylor & Francis, **2002**, 40, 745-760.