



**UNIVERSIDAD NACIONAL AUTÓNOMA  
DE MÉXICO**

---

---

**FACULTAD DE CIENCIAS**

**MODELO DE TRÁNSITO PERIÓDICO VEHICULAR URBANO.**

**T E S I S**

**QUE PARA OBTENER EL TÍTULO DE:**

**MATEMÁTICO**

**P R E S E N T A:**

**EMILIANO CRUZ LEÓN**

**DIRECTOR DE TESIS:**

**DR. RICARDO GÓMEZ AÍZA**



**2009**



Universidad Nacional  
Autónoma de México

Dirección General de Bibliotecas de la UNAM

**Biblioteca Central**



**UNAM – Dirección General de Bibliotecas**  
**Tesis Digitales**  
**Restricciones de uso**

**DERECHOS RESERVADOS ©**  
**PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.



Universidad Nacional  
Autónoma de México



**UNAM – Dirección General de Bibliotecas**  
**Tesis Digitales**  
**Restricciones de uso**

**DERECHOS RESERVADOS ©**  
**PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

# Índice general

<b>Introducción</b>	<b>VII</b>
<b>1. El transporte y el tránsito periódico vehicular urbano</b>	<b>1</b>
1.1. Variables principales del transporte . . . . .	2
1.2. Equilibrio . . . . .	6
1.3. Tránsito periódico vehicular urbano . . . . .	8
<b>2. El trabajo sobre el tránsito</b>	<b>9</b>
2.1. Demanda . . . . .	10
2.1.1. Acercamiento tradicional . . . . .	11
2.1.2. Acercamiento de consumidor . . . . .	12
2.2. Variables clásicas de flujo . . . . .	12
2.3. Modelos de asignación de flujo . . . . .	13
2.3.1. Modelos estocásticos . . . . .	14
2.3.2. Manual de capacidad de carreteras . . . . .	15

<b>3. Descripción del modelo</b>	<b>17</b>
3.1. La representación . . . . .	17
3.1.1. Datos de entrada . . . . .	17
3.1.2. Capas de usuario . . . . .	23
3.2. El problema . . . . .	24
3.2.1. Heurísticamente . . . . .	25
3.2.2. Formalmente . . . . .	25
3.3. Descripción de una jornada . . . . .	26
3.3.1. Seguimiento de vehículos . . . . .	27
3.3.2. Criterios de selección de ruta . . . . .	31
3.3.3. Inicio y término de una jornada . . . . .	36
3.4. Iteración de jornadas. Convergencia . . . . .	37
3.5. Resultados . . . . .	38
3.5.1. Niveles de servicio . . . . .	38
3.5.2. Consumo de energía y emisión de contaminantes . . . . .	39
3.5.3. Índice de correlación de vías . . . . .	41
3.5.4. Tiempos de viaje y ruteo . . . . .	42
3.6. Justificación . . . . .	42
3.6.1. Término de jornada . . . . .	42
3.6.2. Preferencias de usuario . . . . .	43
3.6.3. Tiempo de recorrido continuo, $TRC(a, t)$ . . . . .	46

3.6.4. Función de demora, $\chi(a, t)$ . . . . .	47
3.6.5. Tiempo de recorrido, $TR(a, t)$ . . . . .	48
3.6.6. Congestión, $\rho(v, t)$ . . . . .	49
3.6.7. Intensidad de paso, $\iota(a, t)$ . . . . .	50
3.6.8. Distancia relativa $\widehat{d}_y^\tau$ y gráfica relativa $\widehat{G}_\tau$ . . . . .	50
3.6.9. Memoria de tiempo de recorrido $MTR$ . . . . .	51
3.6.10. Situación, $S_\tau(v, t)$ . . . . .	52
3.6.11. Situación futura, $S_\tau^*(w, b, t)$ . . . . .	53
3.6.12. Efectividad, $E_\tau(b, t)$ . . . . .	53
3.6.13. Asignación, $A_\tau(v, t)$ . . . . .	54
3.6.14. Estabilidad, $ES(n)$ . . . . .	56
<b>4. Diseño del Simulador</b> . . . . .	<b>57</b>
4.1. Convenciones del pseudo-código: . . . . .	58
4.2. Datos de entrada . . . . .	61
4.3. Objetos . . . . .	61
4.4. Algoritmo . . . . .	65
4.4.1. CicloGeneral( $G, \pi, \Lambda, \epsilon, \sigma, N$ ) . . . . .	67
4.4.2. JORNADA( $G, n$ ) . . . . .	70
4.5. Métodos . . . . .	73
4.5.1. CONSTRUYE- $N(v)(G)$ . . . . .	74

4.5.2.	CALCULA- $\rho(v)(t)$ . . . . .	75
4.5.3.	ASIGNA-DESTINO( $v$ ) . . . . .	76
4.5.4.	SELECCIONA- $VEH(v)()$ . . . . .	78
4.5.5.	MONITOREO- $M(a)(\tau, t)$ . . . . .	80
4.5.6.	CALCULA- $TR(a)(t)$ . . . . .	82
4.5.7.	CALCULA- $E(a)(\tau, t)$ . . . . .	83
4.5.8.	CALCULA- $P_\tau(\tau)()$ . . . . .	84
4.6.	Procedimientos . . . . .	85
4.6.1.	Borrar( <i>índice</i> , <i>Lista</i> []) . . . . .	85
4.6.2.	Buscar( <i>elemento</i> , <i>Lista</i> []) . . . . .	86
4.6.3.	BuscarBorrar( <i>elemento</i> , <i>Lista</i> []) . . . . .	86
4.6.4.	UniónConj( <i>Lista1</i> , <i>Lista2</i> , <i>ListaUnión</i> []) . . . . .	86
4.6.5.	VolteaG( $G$ ) . . . . .	87
4.6.6.	Calcula- $\iota(v)$ . . . . .	88
4.6.7.	Completa- $MTR$ . . . . .	89
4.6.8.	Construye- $d_y^\tau(G)$ . . . . .	90
4.6.9.	Construye- $\widehat{d}_y^\tau(G)$ . . . . .	92
4.6.10.	Construye $G_\gamma(G, \gamma)$ . . . . .	94
4.6.11.	Dijkstra, DJT( $G, x, f$ ) . . . . .	96
4.6.12.	Dijkstra dinámico, DJTdin( $G, \tau, f, y, t_0$ ) . . . . .	98
4.6.13.	Dijkstra escalonado, DJTdin( $G, \tau, c, tr, y, t_0$ ) . . . . .	100

---

4.6.14. EvalConv( $MF[\square\square], N, \epsilon, G, n$ ) . . . . .	102
4.7. Procedimientos especiales . . . . .	104
4.7.1. NivelDeServicio( $G, MTR^*(\square), TAB[\square], \eta$ ) . . . . .	104
4.7.2. EnergíaYcontaminantes . . . . .	105
4.7.3. CorrelaciónDeVías(a) . . . . .	107
<b>A. Dijkstra</b>	<b>109</b>
A.1. Problema clásico . . . . .	110
A.1.1. Dijkstra clásico . . . . .	110
A.1.2. Distancias a flujo libre, $d_x^r, d_y^r$ . . . . .	112
A.2. Problema dinámico . . . . .	113
A.2.1. Condición de fila . . . . .	114
A.2.2. Dijkstra dinámico . . . . .	117
A.2.3. Distancia dinámica $d_y^r(v, t)$ . . . . .	122
A.3. Problema relativo . . . . .	126
A.3.1. Dijkstra escalonado . . . . .	127
A.3.2. Distancia relativa $\widehat{d}_y^r$ . . . . .	129
<b>Glosario de Símbolos</b>	<b>131</b>



# Introducción

Hoy en día la mayor parte de la población mundial habita en ciudades y su vida cotidiana se encuentra a merced de la dinámica de éstas. El tránsito nos habla de la calidad y cualidad de vida de una urbe y sus habitantes; de la relación de éstos con su entorno, del espacio que habitan. Nos interesa aportar desde lo que sabemos y hacer matemáticas en el intento.

El trabajo que presentamos dota de una estructura mesoscópica al fenómeno del tránsito, sobre la cual se construye un algoritmo que simula el desarrollo del flujo respecto al tiempo. La estructura que proponemos se basa en una gráfica dirigida con funciones sobre sus flechas y vértices que representa la red vial y las características de sus cruceros y vías. Incorporamos a los conductores y sus vehículos como flujo sobre las flechas. Las características de los vehículos, conductores y viajes a realizar determinan diferentes tipos de usuario y nos permiten distinguir, dentro del flujo global, flujos parciales que corresponden a los distintos tipos de usuario. Finalmente, el motor que mueve la estructura son las necesidades de viaje que incorporamos como datos de *origen-destino*.

El primer capítulo presenta el tránsito periódico en el marco de los sistemas de transporte. Consideramos las tres variables principales del transporte en cuyas características y relaciones se puede descomponer el fenómeno del tránsito. Aterrizando los conceptos, se identifican estas características y sus relaciones respecto a la red vial, sus usuarios, gestores y los volúmenes de vehículos que circulan en ella.

El transporte, como sistema complejo, puede verse como la interacción de varios subsistemas que funcionan cualitativamente distinto aunque de manera conjunta, por lo que diferentes tipos de acercamiento pueden requerirse para cada subsistema. El segundo capítulo expone una gama básica de acercamientos para modelar el tránsito, en términos de los objetivos que persiguen y las relaciones y características del tránsito que consideran. Los acercamientos contemplados están presentes en las distintas secciones que componen el modelo. Respecto al tránsito hay mucho trabajo realizado desde muchas disciplinas, el modelo que proponemos retoma un conjunto de conceptos y métodos incorporándolos en una estructura que los reúna y permita un acercamiento dinámico y mesoscópico al problema del tránsito.

Consideramos las calles y cruces como los actores principales del modelo, tratando las propiedades globales de la red como características emergentes. Una de estas características es la distancia, que definimos de manera clásica en el contexto de la teoría de gráficas mediante el costo de los caminos mínimos, en la gráfica dirigida que representa la red vial, utilizando como función de costo en las flechas el tiempo que toma recorrerlas. Cualquier otra variable del modelo se basa en esta distancia y en las características locales respecto a las flechas y vértices. De esta manera establecemos una fuerte relación entre los resultados del modelo y la estructura de la gráfica, con la intención de, a posteriori, recuperar información estructural de la gráfica mediante los flujos que induzca su estructura y determinar características del flujo mediante características combinatorias de la gráfica.

Tomamos en cuenta distintos tipos de usuario, sus percepciones y preferencias sobre la red y sus características: seguridad, comodidad, entorno... Estas consideraciones repercuten en la noción de distancia de los usuarios, lo que modifica la geometría de la red. Más aún podemos considerar una calle o vía *fuera de la red*, en caso de que un conductor no considere prudente atravesarla, lo que modifica la topología de ésta en función de la percepción del usuario. Esta transformación de la red respecto al tiempo y la relación con sus usuarios evidencia una propiedad emergente del tránsito en que las ciudades reaccionan dilatándose, contrayéndose, quebrándose y volviéndose a unir en respuesta a las actividades de quienes las habitan y los acontecimientos que en ellas suceden.

El trabajo concluye con el pseudocódigo de un algoritmo que, basado en el modelo, encuentra una sucesión de flujos, parametrizada por el tiempo, que representa los vehículos moviéndose en la red. Llamamos *jornada* a dicha sucesión, durante la cual se guardan ciertos datos que funcionan como memoria de los usuarios. El proceso se repite incorporando la memoria de la *jornada* anterior para obtener una super-sucesión de *jornadas* cuya convergencia se evalúa hasta encontrar una razonablemente estable. Es un algoritmo de tiempo de ejecución polinomial.

El resultado del trabajo no es un simulador de tránsito vehicular que pueda directamente aplicarse a una situación real. El trabajo necesario para aplicar el modelo requiere la acción de varias disciplinas así como de trabajo de campo. Por tales razones, la aplicación queda fuera de los alcances de esta tesis. Nos limitamos a formular un modelo que permita desarrollar dicha aplicación, un modelo cuya estructura se apegue lo más posible a las características del fenómeno mesoscópico del tránsito y a la vez permita el cálculo, predicción y estudio de flujos vehiculares y la red vial mediante el estudio de su interacción dinámica.

Como apéndice incluimos dos algoritmos generalizados del construido por Dijkstra. El que denominamos *Algoritmo Dinámico de Dijkstra* resuelve el problema del camino más corto cuando el tiempo de recorrido (o la longitud) de las flechas es dependiente del tiempo (en que se inicia el recorrido). Esta función dinámica de tiempo de recorrido es sensible a la situación del flujo en cada tiempo, de modo que considera la congestión a lo largo de la jornada. El *algoritmo relativo de Dijkstra* es en realidad una aplicación de la versión clásica a una gráfica estratégicamente modificada de la original. Esta versión encuentra el camino de costo mínimo para una función de costo que

depende del tiempo pero no es forzosamente el tiempo de recorrido.

La parte que más me interesa y más disfruté de esta tesis es el acercamiento al fenómeno, el articular lo técnico con lo subjetivo y casi personal, las percepciones con las distancias, lo aleatorio dentro del determinismo... El principal resultado de esta tesis es un modelo de asignación de flujo (que me gusta pensar) apegado al fenómeno del tránsito, que admite (como se muestra en el capítulo 4) un algoritmo de asignación cuyo tiempo de ejecución es de orden polinomial. Este algoritmo tiene como entradas principales la red vial y los datos de origen-destino de los viajes que periódicamente se llevan a cabo en ella y da como resultado, cada cierto tiempo durante un lapso general de tiempo, el número de vehículos que se encuentran en cada una de las vías de la red.

# Capítulo 1

## El transporte y el tránsito periódico vehicular urbano

El transporte es la acción de llevar algo de un lugar a otro. El polen arrastrado por el viento, los alimentos recolectados por las hormigas, la luz viajando en el espacio, el oxígeno en el torrente sanguíneo, los electrones en los cables de luz, la migración de la ballena gris. Es difícil pensar en una actividad que no requiera de transporte, sobre todo si involucra varios actores.

Dentro de toda la gama de fenómenos de transporte distinguimos primero aquellos relativos a la civilización: el correo, electrónico o clásico; el transporte de agua potable, de basura, de pasajeros, de mercancía; la telefonía, el periódico, la tradición oral; y en general los acontecimientos que responden a necesidades sociales y suceden en medios administrados o creados por la sociedad. Dentro de este universo definimos un *sistema de transporte* como la unión dinámica de dos subsistemas: Un sistema de elementos físicos manejados por organizaciones humanas para mover personas y bienes, llamado *infraestructura de transporte*; y el sistema formado por las fuerzas sociales, económicas y políticas en relación al transporte, denominado *sistema de actividades*<sup>1</sup>. Si bien la definición de transporte podría abarcar una gama más amplia de fenómenos, esta definición basta para los alcances de este trabajo.

El sistema de transporte vehicular urbano es aquel relativo a una zona urbana y formado por la red vial, sus características y reglamentos; que se lleva a cabo esencialmente por automóviles, camiones (de pasajeros y carga) y motocicletas, y que responde a las actividades creadas por las diversas necesidades sociales del lugar.

---

<sup>1</sup>Aunque nuestra definición diverge de la del autor, se gestó a partir de los conceptos expuestos en [MM] página 14

## 1.1. Variables principales del transporte

Un sistema es un conjunto de elementos relacionados entre sí y encaminados hacia ciertos objetivos específicos y metas. El sistema de transporte se encarga de proveer el transporte necesario para las distintas actividades que lo requieren. En términos de sus elementos se caracteriza mediante tres variables principales: la infraestructura de transporte, el sistema de actividades y el patrón de flujos<sup>2</sup>.

El *flujo* consiste de los volúmenes de bienes y/o gente moviéndose a través del sistema. Cuando hablamos de transporte vehicular el flujo se refiere al volumen de vehículos del sistema. Las características del flujo recaen en su composición: tipos de vehículo (compactos, camiones, autobuses), orígenes-destinos, tipos de conductor (edad, habilidades, velocidad de manejo) y tipos de viaje (habitual o de recreo).

La infraestructura de transporte se refiere a todos los elementos físicos que colaboran en el transporte, lo que incluye las arterias (calles, avenidas, distribuidores, pasos a desnivel...), lugares de estacionamiento no arterial, señales, semáforos, reglamentos, parque vehicular y tecnología de monitoreo y regulación.

Las características de infraestructura se clasifican de acuerdo a sus elementos:

Elementos de Infraestructura y sus Características <sup>3</sup>	
Arterias	Diseño geométrico: ancho de carriles, sobreelevación o acotamiento, cunetas, alineación vertical y horizontal, superficie de rodamiento. Función proyectada: tipo de vía, entorno urbanístico, velocidad de proyecto, lugares que une, capacidad de proyecto.
Estacionamiento	Ubicación, capacidad, costo y horario de servicio.
Señales	Espaciamiento, diseño y tipo de señal.
Semáforos	Tipo de semáforo: tiempo fijo, semi accionado por el flujo (según horario), accionado por el flujo. Duración del ciclo semafórico Calidad de la progresión: tiempo efectivo de verde por maniobra. Ubicación (respecto a otros semáforos).
Reglamento	Preferencias en intersecciones, vueltas continuas, velocidades máximas y mínimas.
Parque vehicular	Velocidad, capacidad de aceleración y frenado, emisión de contaminantes, tipo y cantidad de energía que consume, capacidad, tamaño, capacidad de giro, seguridad, peso.
Tecnología	Alcances, control dinámico de flujo, cobertura, función.

<sup>2</sup>[MM] página 12

El sistema de actividades comprende a los usuarios, sus necesidades y patrones de comportamiento; los métodos por los cuales actúan las instituciones sociales encargadas del tránsito o cualquier organización social que sea capaz de incidir en él. Esencialmente se refiere a la dinámica por la cual la sociedad incide en el sistema de transporte.

Elementos del Sistema de Actividades y sus Características	
Usuarios	Origen-destino-horario de sus actividades Modo de transporte Preferencias: seguridad, entorno urbano, caminos principales y velocidad.
Instituciones u Organizaciones sociales	Tiempo de reacción: evaluación, solución e implementación Capacidad de reacción: coordinación efectiva, recursos. Alcances de solución: corto, mediano, largo plazos Prioridades: ecológica, económica, social, política.

Existen tres relaciones esenciales de estas variables<sup>4</sup>:

1. El patrón de flujos está determinado por la infraestructura de transporte y el sistema de actividades.
2. El patrón de flujos, con el tiempo, incidirá en el sistema de actividades a través de su influencia en los patrones de servicios de transporte ofrecidos y los recursos consumidos en brindar dichos servicios.
3. El patrón de flujos, con el tiempo, incidirá en la infraestructura de transporte mediante la acción del gobierno que, en respuesta a flujos presentes o previstos, desarrollará o modificará la infraestructura de transporte con miras a mejorar el patrón de flujos.

La primera relación asevera que el patrón de flujos es una repercusión inmediata de la infraestructura de transporte y el sistema de actividades. De hecho, el patrón de flujos no es parte del sistema de transporte sino consecuencia de éste; la variable característica del tránsito producido por dicho sistema.

El sistema de actividades incide en el flujo mediante los usuarios y sus preferencias, pues son sus necesidades de transporte las que generan los viajes y determinan la elección de las rutas de acuerdo con las diferentes características de éstas. Los efectos del sistema de actividades sobre el flujo son:

---

<sup>4</sup>[MM] página 13

**Tipo de vehículo:** Los vehículos pesados son más grandes que los de tamaño estándar, ocupan más espacio dentro de las arterias; por otro lado las capacidades de movimiento (aceleración y frenado, velocidad en pendientes ascendentes) son menores por lo que requieren de infraestructura especial (pendientes pequeñas o cortas, mayor radio de giro) para no obstaculizar la circulación.

**Tipo de conductor:** Las distintas características (capacidades físicas, conocimiento de la vía, hábitos de manejo) afectan el desplazamiento de los vehículos en una vía y por lo tanto el flujo dentro de ella. Por ejemplo, los conductores habituados a la vía realizan las maniobras necesarias en los momentos precisos optimizando su recorrido por ella, los conductores no habituales, por el contrario, se desenvuelven con menor precisión ocasionando demoras innecesarias para ellos mismos y otros vehículos. Conductores de distintas edades tienen diferentes capacidades de reacción a una señal de alto o un accidente, y suelen manejar a distintas velocidades en la misma arteria<sup>5</sup>.

**Origen-Destino:** El origen y destino del trayecto de un conductor determina las maniobras que éste realiza dentro de la arteria, las cuales afectan el tránsito de distintas maneras dependiendo de su ingreso en la vía y su salida de ella. Un vehículo que entra a una vía por el carril izquierdo y requiere dar vuelta a la derecha al final de éste, tiene que atravesar todos los carriles; a diferencia de uno que entra y sale recto.

Por su parte, la infraestructura de transporte incide en el flujo de varias formas:

**Diseño geométrico de arterias:** proporcionan seguridad al conductor para ir a la velocidad que desea; un diseño errado (carriles angostos obstáculos en los costados de la vía como bardas, postes, alcantarillas o canales) obliga al conductor a ir despacio para obtener mayor control sobre el vehículo y evitar accidentes.

**Alineación vertical u horizontal:** Se refiere a las subidas, bajadas así como al peralte (sobreelevación), las primeras afectan la velocidad de los vehículos, ya sea por mayor dificultad en las subidas o por precaución en bajadas. Por su parte la existencia de peralte adecuado permite velocidades más altas, al contrario del peralte inadecuado que requiere precaución y/o disminución de la velocidad para no perder el control del vehículo.

**Superficie de rodamiento:** La presencia de baches y otros defectos en la superficie de rodamiento aumenta la probabilidad de accidentes y daña los vehículos, lo que obliga a los conductores a reducir su velocidad para evitar dichas desavenencias.

**Tipo de vía:** La función de todas las vías es proveer de acceso y movilidad a los vehículos de manera que puedan salir de su orígenes trasladarse y alcanzar su destino. Se clasifican en vías primarias, secundarias y locales de acuerdo a la proporción en que proveen movilidad y accesibilidad. Las vías locales se encargan principalmente del acceso, la movilidad que ofrecen se limita a conectar

---

<sup>5</sup>[MCC] página 85.

los orígenes y destinos con las vías secundarias y primarias. Del otro lado, las vías primarias están diseñadas para proveer movilidad lo que muchas veces reduce el acceso que brindan que llega a ser nulo en vías de acceso restringido. Las vías secundarias están en un nivel intermedio. En un viaje típico, los vehículos salen de su origen a una vía local que los conduce a la red de vías primarias y secundarias por la cual se acercan a su destino para finalmente tomar una vía local que les permite alcanzarlo.

Las vías primarias son acceso total o parcialmente restringido (periférico, circuito interior, viaducto, etc), con velocidades permitidas más altas y de circulación total o parcialmente continua (ya sea con preferencia de paso de paso en los cruces y la sincronización y progresión semafórica a su favor; o con rampas de entrada y salida y pasos a desnivel que eviten cruces y cortes a la circulación) en su mayor parte. Las vías secundarias son de circulación interrumpida de acceso libre y velocidades permitidas medias. En las vías locales es común que los vehículos se detengan para estacionarse, las velocidades permitidas son bajas, suelen presentar distintas dificultades para la circulación como doble sentido sin separación física o el ser más angostas.

El tipo de vía determina la fluidez de proyecto de la vía mediante su relación jerárquica con otras vías, lo que se traduce en los criterios de preferencia en las intersecciones así como los mecanismos de incorporación y salida de vehículos de la vía.

**Entorno urbanístico:** Un entorno de alta densidad comercial o residencial obliga al conductor a reducir la velocidad. Una zona insegura suele ser evitada por la mayoría de los conductores aunque signifique un rodeo mayor, al contrario de una zona agradable que puede atraer conductores aunque implique un camino más largo.

**Velocidad de proyecto:** Es la velocidad a la que el planeador considera se recorrerá la vía. La distancia y tamaño de las señales así como la sincronización de los semáforos y el diseño geométrico de la vía dependen de ella. Para un vehículo que viaja por arriba (debajo) de dicha velocidad los peraltes y radios de las curvas pueden resultar peligrosos (excesivos, más recorrido del necesario para una vuelta), las señales poco (muy) anticipadas y encontrará luces en rojo a menudo durante su recorrido. A la velocidad de proyecto las señales avisan a tiempo y la coordinación de semáforos es la mejor posible de forma que se encuentren la mayoría en verde.

**Capacidad de proyecto:** Es el flujo máximo, según el proyecto, para el que la arteria sigue prestando el servicio planeado dentro de las condiciones proyectadas (como la velocidad de proyecto). Es decir, por arriba de la capacidad de proyecto se espera que la vía en cuestión se congestione lo que implica una restricción para el flujo.

**Configuración de carriles:** Tipo de carriles, su diseño en el espacio, por ejemplo: carril especial de vuelta izquierda, de retorno, carriles para estacionamiento. Determinan el espacio para las distintas operaciones o maniobras dentro de la arteria. Una arteria con pocos espacios ocasiona conflictos entre los conductores para poder realizar cada uno la maniobra que requieren.

**Diseño de red:** Dado que el motivo de los viajes es trasladarse de un lugar a otro, el flujo de una arteria o vía depende de los lugares que une, se conforma de vehículos para cuyo viaje sea de utilidad cambiar de posición, del lugar inicial de la vía o arteria a su lugar final. En este sentido la red en su totalidad, tomando en cuenta todas las aristas, ofrece un conjunto de opciones a quienes la utilizan para transportarse, el flujo se desenvuelve dentro de este conjunto.

Además el flujo actúa sobre sí mismo. Cuando el flujo es grande las distancias entre los vehículos disminuyen, provocando que los conductores reduzcan su velocidad para mantener cierta distancia de seguridad con el vehículo que les antecede. Hay un efecto similar en intersecciones sobrecargadas: cuando muchos vehículos quieren cruzar se produce un conflicto del que resultan colas en las vías de acceso a la intersección, sometiendo a la circulación a un proceso de desahogo.

La segunda y tercera relaciones tratan de cómo, con el paso del tiempo e indirectamente, el patrón de flujos afecta, en una retroalimentación, la infraestructura de transporte y el sistema de actividades.

Cuando la capacidad de las vías principales entre un origen-destino es superada, el flujo busca distintos caminos, ya sea utilizando las vías secundarias (cuya función es de acceso y no de paso), cambiando su función y, por lo tanto, obstruyendo el servicio para el que están construidas; o cambiando el sistema de actividades, saliendo antes (para absorber el tiempo perdido en la congestión, evitarla o una combinación de ambas), llegando después, cambiando el horario, vehículo, lugar o día de la actividad a realizar.

En un segundo momento, las instituciones u organizaciones relacionadas con el transporte reaccionan con estrategias para resolver los problemas o atender las nuevas necesidades de transporte. Éstas pueden tener distintos objetivos que repercutirán ya sea en la infraestructura de transporte (mejoras o extensiones de la red, incorporación y mejoramiento de otros servicios o tecnologías de transporte, reorganización del reglamento como cobros por entrar a zonas congestionadas, parquímetros etc.) o en el sistema de actividades (políticas de uso del suelo, promoción del transporte público, de más pasajeros por vehículo, uso de bicicletas o modos de transporte alternativo). Ésto completa el ciclo pues los cambios realizados incidirán, de acuerdo a la primera relación, en el patrón de flujos.

## 1.2. Equilibrio

El equilibrio de un sistema es un estado de reposo, una situación en que las fuerzas que actúan sobre él se contrarrestan. Para el sistema de transporte definimos tres tipos o niveles de equilibrio, el equilibrio de viaje, el de sistema y el equilibrio ideal.

El *equilibrio de viaje* se caracteriza en que todos los usuarios están seguros de que su rutina de viajes es la que más les convence, de forma que cualquier cambio de ruta de cualquier usuario significa para él una pérdida. Es un equilibrio donde sólo tomamos en cuenta la primera relación esencial de las variables del sistema; un equilibrio a corto plazo que omite las repercusiones del patrón de flujos en las otras dos variables. Un patrón de flujos correspondiente a un equilibrio de viaje se mantendrá estable (con todo y la congestión que implique) hasta que haya cambios en la infraestructura o en las actividades.

El *equilibrio de sistema* se caracteriza por que todos los usuarios están seguros de que los lugares y horarios en que realizan sus viajes son los que más les convienen; es decir, cualquier cambio en los horarios de salida o rutas de un usuario implica una pérdida para éste. Es un equilibrio a mediano plazo donde se toman en cuenta las primeras dos relaciones esenciales del sistema, un equilibrio que sólo omite las repercusiones a largo plazo del flujo sobre la infraestructura. Un patrón de flujos correspondiente a un equilibrio de sistema se mantendrá estable hasta que haya cambios en la infraestructura o modificaciones mayores en las actividades (más allá del horario de salida).

El *equilibrio ideal* es aquel en que cualquier cambio en el sistema de transporte, o en el uso de éste por algún usuario, implica una pérdida en la eficiencia de su funcionamiento para el usuario. Es un equilibrio a muy largo plazo, ideal pues supone que las necesidades sociales de transporte no cambian. Un patrón de flujos correspondiente a un equilibrio ideal se mantendrá siempre estable.

De acuerdo a la dinámica del sistema hay diferentes tipos de equilibrio. Los equilibrios puntuales son, como los descritos anteriormente, los que no cambian en el tiempo. Existen equilibrios cíclicos, que cambian pero mantienen siempre un ciclo, como la tierra alrededor del sol (no está en equilibrio puntual pues cambia de posición, sin embargo cada año realiza el mismo recorrido respecto al sol). Los equilibrios extraños o caóticos, que no cambian mucho pero si constante e impredeciblemente, como el amor.

Finalmente hay equilibrios estables e inestables. Los estables son aquellos que no se alteran por ligeros o grandes cambios de las condiciones del sistema, como la burocracia o las necesidades fisiológicas. Los equilibrios inestables son aquellos que se pierden por pequeños o mínimos cambios en el sistema, como el buen humor.

El estudio de equilibrios es parte esencial del estudio de un sistema pues nos dice los posibles escenarios de éste según los cambios esperados o aquellos que somos capaces de provocar<sup>6</sup>.

---

<sup>6</sup>[OW] páginas 321-337.

### 1.3. Tránsito periódico vehicular urbano

Por *tránsito* entendemos el transporte de personas o bienes por una red. El *tránsito vehicular urbano* consiste entonces de los vehículos moviéndose por la red vial. Nos interesa conocer su dinámica y repercusiones: los embotellamientos, cuellos de botella, colas de espera, velocidades y tiempos de viaje, accidentes, gasto de energía, producción de contaminantes, generación de estrés. . . es decir, todo aquello que nos diga cómo, en qué condiciones y con qué consecuencias se desarrolló el tránsito. En este trabajo no estamos considerando qué es lo que transportan los vehículos, sino los vehículos en sí mismos; un autobús del sistema de transporte público es para el modelo un vehículo más (a pesar de su mayor eficiencia, frente al automóvil, en el transporte de personas).

El *tránsito periódico* es el desarrollo de los fenómenos de transporte que se repiten con cierta regularidad. Es como los clientes frecuentes de un bar, un día normal de escuela o la rutina de un trabajador. Antagónico a él es el tránsito aleatorio: el niño que se coló por una cerveza, el día de la excursión en la primaria, el día festivo del trabajador. La periodicidad nos permite establecer un comportamiento base del tránsito, sobre el cual suceden los eventos aleatorios. El tránsito periódico es estable, es lo mínimo que el sistema de transporte debe ser capaz de sostener. El tránsito aperiódico representa las situaciones emergentes a las que el sistema debe responder cuando ocurran.

## Capítulo 2

# El trabajo sobre el tránsito

El sistema de transporte tiene similitud natural con la teoría microeconómica si lo entendemos como un servicio para usuarios prestado por los administradores de la infraestructura de transporte. Lo mismo sucede con el transporte periódico vehicular urbano si restringimos usuarios a conductores (exactamente uno por vehículo) y consideramos solamente a los administradores relacionados con la red vial.

Llamamos *demanda de transporte* al conjunto de todas las decisiones individuales acerca de los viajes. Así mismo, denominamos usuarios (conductores) a aquellos que ejercen dicha demanda haciendo uso de la red vial. La demanda proviene del sistema de actividades, pues son los usuarios los que toman las decisiones de los viajes. El efecto inmediato de las instituciones u organizaciones sociales sobre la demanda se expresa a través de los usuarios. La oferta la forman todos aquellos servicios de transporte enfocados a satisfacer la demanda de los usuarios; corresponde a la infraestructura de transporte.

“En términos económicos convencionales, el intercambio de bienes y servicios es el resultado de la combinación de la demanda y la oferta de ellos. El punto de equilibrio resultante de esta combinación define el precio al que los bienes serán intercambiados y sus respectivos flujos en el mercado.”<sup>1</sup> De esta forma, el tránsito es el intercambio de transporte por los diferentes costos que implica para los usuarios (estrés, tiempo, combustible, impuestos, mantenimiento y adquisición del vehículo).

Las dos partes esenciales en las que se divide el estudio del tránsito son los estudios de demanda y de flujos. El objetivo principal de los estudios es la planeación; evaluar la situación en que se encuentra el tránsito y como intervenir para mejorarla.

---

<sup>1</sup>[OW] página 321

Se consideran dos niveles de evaluación, el microscópico y el macroscópico. La diferencia entre ambos es la proporción que guardan sus actores directrices con el objeto de estudio. Los actores principales son aquellos que sustentan o dirigen la evaluación. Una evaluación microscópica del sistema de transporte utiliza actores relativamente pequeños respecto a él, como los vehículos, de los que se mide *espaciamiento inter-vehicular, radios de giro, espacios de giro, espacio de espera, tiempo de reacción*. . . Una evaluación macroscópica se basa, por ejemplo, en rutas completas para las que se toman en cuenta variables como la *longitud* y el *tiempo de recorrido*. Por último, en un nivel intermedio, la evaluación mesoscópica tiene como sub-actores aquellos de la evaluación microscópica y como super-actores a los de la macroscópica, por ejemplo los cruceros o arterias en las que se utilizan variables como la *composición y magnitud del flujo, capacidad del cruce y preferencia para giros*.

Para efectos prácticos el enfoque microscópico consiste de análisis detallados de zonas pequeñas de la red (casi siempre una vía o cruce); el macroscópico se refiere al análisis de zonas que consideran varias vías y cruceros, o al análisis de la red completa. Sin embargo ambos niveles pueden realizar análisis de la red completa o de un cruce o arteria específicos. Cada enfoque utiliza y proporciona un tipo distinguido de datos e información cuyas características correspondan a los actores principales y los métodos propios de la evaluación. Los resultados obtenidos corresponden, entonces, a las características contempladas de los actores así como a su papel dentro del fenómeno.

## 2.1. Demanda

A nivel microscópico la demanda consiste, en el caso de un cruce, del número de vehículos que realizan cada una de las posibles operaciones dentro de él, a saber, vuelta izquierda, derecha, cruzar recto y vuelta en “U”, para cada calle que entra al cruce (para un cruce estándar entre dos vías); en el caso de una vía la demanda es el número de vehículos que la cruzan. Los estudios clásicos que se encargan de determinar la demanda a nivel microscópico se realizan mediante observaciones: conteos vehiculares para cada operación posible de un cruce, en su caso; y conteos de los vehículos que cruzan una calle.

En el nivel macroscópico la demanda consiste del número de vehículos que van de una región a otra dentro de la zona de estudio; esta zona de estudio puede ser una parte de la red o la red completa. Los estudios clásicos que se encargan de determinar la demanda, a nivel macroscópico, se realizan mediante diferentes tipos de encuestas origen-destino.

### 2.1.1. Acercamiento tradicional

Es conocido como el Acercamiento de los Cuatro Pasos: generación de viajes, distribución de viajes, divergencia modal y asignación de viajes. Los primeros tres son una división de la demanda y el cuarto se refiere al flujo. En principio este acercamiento pretende que los cuatro pasos se lleven a cabo secuencialmente de forma que cada uno utilice los resultados del anterior. Finalmente el último paso da como resultado un patrón de flujos (generalmente el asociado al equilibrio de viaje)<sup>2</sup>.

Todo modelo global de tránsito tiene que resolver las cuatro fases que describe el acercamiento tradicional aunque no lo haga secuencial e independientemente, como el acercamiento ortodoxo sugiere. Los pasos que conforman el procedimiento para determinar la demanda (primeros tres) son los siguientes:

1. Generación de viajes:

Tiene como propósito estimar el número y tipo de viajes que generan (atraen) cada uno de los vértices o zonas de la gráfica. Consiste en determinar los viajes que genera una zona de acuerdo a las actividades que se realizan en ella.

Los modelos clásicos de generación se basan en las características económicas de la zona (zona habitacional, comercial, industrial) y de sus ocupantes. Por ejemplo tomemos una unidad habitacional; en la mañana genera los viajes hacia los trabajos y las escuelas absorbiendo pocos viajes; a media tarde absorbe los viajes de regreso y genera los de las actividades de recreación, las talachas, el abastecimiento de alimentos... En este paso no es importante a qué lugar en particular se generan los viajes, sino cuántos se generan, de qué tipo y en qué horario (56 niños salen a la escuela a las 8:00 am; Procter & Gamble absorbe los viajes de 100 trabajadores a las 9:00 am).

2. Distribución de viajes:

Consiste en distribuir los viajes originados en cada vértice junto con los viajes finalizados en cada vértice, es decir, asociar a cada viaje un origen y un destino de los disponibles por el proceso anterior (generación de viajes). Siguiendo con nuestro ejemplo, al grupo de viajes generados por los trabajadores que viven en la unidad habitacional se les distribuye en los puntos que atraen trabajadores. Así se obtiene parcialmente la información de origen-destino (20 niños de la unidad habitacional van a la escuela Primaria No.1 a las 8:00 am)

3. Divergencia modal:

Reparte los viajes en los distintos modos (automóvil, transporte público, peatonal, bicicleta, etc.) según su origen, destino, situación económica, etc. Dicho de otra forma, ya que sabemos quienes van y de dónde a dónde van (por el paso anterior), en este paso se determina cómo lo

---

<sup>2</sup>Como referencia consultar [OW] capítulos 4,5,6,7; [WS] páginas 11-20; [MM] página 32

hacen, es decir, en qué medio de transporte (5 niños de la unidad habitacional van a la escuela No.1 en auto, 10 en transporte público, 5 en bicicleta, 3 en helicóptero y 2 caminando).

### 2.1.2. Acercamiento de consumidor

Ataca el problema de determinar la demanda utilizando un esquema económico de maximización de la utilidad o beneficio. La decisión final de un usuario de hacer un viaje implica un procedimiento que toma en cuenta las restricciones (económicas, accesibilidad territorial, acceso a vehículo, horario, etc.), oportunidades (distintas opciones) y objetivos del usuario (beneficios de las actividades a realizar). Este procedimiento de decisión se supone racional en el sentido de obedecer a la maximización de beneficios para el usuario.

Se asigna un costo y beneficio a cada una de las opciones del usuario (modo de transporte, horario de salida, destinos), así como a la actividad asociada al viaje (viajes forzados como la escuela y el trabajo, u opcionales como los de recreación y compras). Finalmente la demanda se compone de las decisiones que maximizan el beneficio del usuario.

La ventaja principal de este acercamiento es considerar juntas todas las variables de la demanda sin otorgar ninguna ventaja a priori a alguna de ellas. Permite incorporar el comportamiento de los usuarios asignando diferentes costos dependiendo de las preferencias y situación de cada uno. Incluso es posible tomar en cuenta el papel de los proveedores de transporte (público o privado) suponiendo que buscan alcanzar objetivos específicos (ganancia, bienestar social) o cumplir con ciertas restricciones.

Es una alternativa a los acercamientos descriptivos (estadísticos por ejemplo) que produce resultados dotados de una explicación (qué opciones elige cada usuario y porque), misma que se encuentra ausente en los modelos descriptivos.

## 2.2. Variables clásicas de flujo

En el nivel microscópico, las magnitudes que comúnmente se utilizan para medir las condiciones del tránsito son:

**Volumen:** Número de vehículos por unidad de tiempo. Se habla de volumen de arterias generalmente, pero aplica también para intersecciones. Nos dice a cuantos vehículos transporta cada arteria o cruce.

**Velocidad:** Tasa de movimiento expresada como distancia recorrida por unidad de tiempo. Se

utiliza un valor representativo dado que en una corriente de tránsito suele haber una amplia gama de velocidades de los diferentes vehículos. Es una medida de la calidad con la que una arteria transporta a un flujo de vehículos, ya que sobre ella actúan muchas variables relacionadas con las características de la arteria y del flujo mismo.

**Densidad:** Número de vehículos por unidad de distancia (1 Km generalmente). Describe la proximidad entre vehículos (espaciamiento) y refleja la libertad de maniobra dentro de la corriente circulatoria.

En el nivel macroscópico, las magnitudes comúnmente utilizadas para describir las condiciones del tránsito son:

**Tiempo de recorrido:** Tiempo que tarda un vehículo en recorrer una segmento o zona establecidos, ya sea una arteria, una ruta o una unidad fija de longitud. Es, análogo a la velocidad, una medida de calidad del transporte en la zona que se especifica.

**Flujo:** Número de vehículos que componen el flujo de una arteria ruta o zona. Mide el número de vehículos a los que una arteria da servicio en un momento preciso.

### 2.3. Modelos de asignación de flujo

El primer problema al hacer un análisis o evaluación del tránsito es determinar qué se considera un sistema eficiente de transporte: los criterios de evaluación. Por supuesto cada evaluador define estos criterios y las variables encargadas de evaluarlos en términos de los objetivos que persigue, lo que se resume en el modelo que utiliza. Sin embargo existe un grupo general de objetivos y características de los modelos de asignación y evaluación del flujo.

Los objetivos principales de la asignación son<sup>3</sup>:

- Obtener medidas agregadas confiables (flujo total o regional por ejemplo).
- Estimar los costos de viaje entre zonas.
- Obtener flujos razonables en las flechas identificando las altamente congestionadas.
- Estimar las rutas utilizadas para cada origen-destino.
- Analizar que origen-destino utiliza una flecha o ruta en particular.

---

<sup>3</sup>[OW] páginas 327-328

- Obtener datos de maniobras de giro para el diseño de futuras intersecciones.

Las entradas básicas son:

- Matriz de viajes que exprese demanda estimada para cada par de nodos.
- Red vial incluyendo propiedades y curvas de velocidad-costo. para sus flechas
- Principios o reglas de selección de rutas.

Cada modelo o método de asignación debe resolver los siguientes problemas en algún momento del proceso.

- Identificación de rutas.
- Asignación de porciones del flujo a dichas rutas o segmentos.
- Búsqueda de convergencia.

### 2.3.1. Modelos estocásticos

Se enfocan en la percepción del costo y la mezcla de valores que los conductores desean minimizar. De esta manera se establece un criterio general de minimización de costos permitiendo variedad en la definición de estos. Los modelos de este tipo con mayor aceptación son los de simulación y los proporcionales<sup>4</sup>.

- Los modelos de simulación asumen para cada vía: un costo objetivo, medido por un observador (modelador); un costo subjetivo, como lo percibe el usuario; existencia de una distribución de costos percibidos por el usuario con el costo objetivo como media. Las distribuciones se consideran independientes. Finalmente se supone que los conductores eligen la ruta que minimiza los costos que perciben.

Es un modelo que respeta la diversidad de comportamientos o preferencias de los usuarios, minimizando para cada uno de ellos el costo total.

---

<sup>4</sup>[OW] paginas 333-335

- Los modelos proporcionales dividen los viajes en las posibles rutas en función de la proporción de los costos de éstas. Las rutas con menores costos cargarán la mayoría de la circulación, lo que eleva su costo. Idealmente, si la función que asigna las proporciones (no forzosamente lineal) es adecuada, los costos de las diferentes rutas se igualarán por la acción del flujo que se les asigna, alcanzando un estado de equilibrio.

Este acercamiento no atiende los costos individuales de los conductores sino un costo colectivo. A diferencia del equilibrio de viaje (ver 1.2), en el equilibrio al que aspiran este tipo de modelos cualquier cambio en la ruta de un usuario implica un aumento (pérdida) del costo (beneficio) colectivo.

### 2.3.2. Manual de capacidad de carreteras

Fue originalmente publicado en 1950 por el entonces Bureau of Public Roads como una guía para el dimensionamiento y análisis de la circulación de las infraestructuras de transporte. El manual vigente actualmente (edición 2000) fue desarrollado por el Transportation Research Board (TRB) Committee on Highway Capacity and Quality of Service.

Los procedimientos de este manual emanan de una amplia gama de investigaciones empíricas desarrolladas desde mediada la década de los sesenta. Los procedimientos reflejan la experiencia operativa en los Estados Unidos, aunque, como en el caso de México, los procedimientos de diseño sugeridos por el manual son ampliamente utilizados con cambios en algunos factores y consideraciones especiales [TM].

Define seis niveles de servicio de una vía o arteria, cuyos criterios de diferencia son la velocidad, el tiempo de recorrido, la libertad de maniobra, las interrupciones a la circulación, el confort y la conveniencia. Las variables que evalúan estos criterios son la velocidad media de recorrido, en el caso de vías de flujo interrumpido, que se calcula como la suma del tiempo de recorrido en movimiento dentro del segmento arterial y la demora total de los movimientos de paso en todas las intersecciones de la vía. En el caso de vías de flujo continuo se utiliza la densidad, ya que refleja con mayor fidelidad que la velocidad las características del tránsito respecto a los criterios de nivel de servicio.

La capacidad de una vía o arteria coincide con el número máximo de vehículos que pueden pasar por un punto dado en un periodo específico respetando el nivel de servicio E. Este nivel de servicio es el último para el que no se produce congestión (el caso del nivel de servicio F)<sup>5</sup>.

Los conceptos y resultados de niveles de servicio de arterias y vías se pueden extender a nivel de servicio de zonas lo que resulta en una evaluación global de la red o alguna de sus zonas. Los métodos del manual se basan en descripciones probabilísticas del tránsito. Esencialmente el

---

<sup>5</sup>[MCC] pagina 6

método consiste en asociarle a la vía un estado ideal, dependiendo de sus características generales (tipo de vía, número de carriles), el cual se modifica mediante el uso de parámetros de acuerdo a las características particulares (ubicación, semáforo, diseño geométrico, giros permitidos, condiciones de la superficie de rodamiento, etc). El resultado final es la capacidad y niveles de servicio de las vías y cruces, junto con las estimaciones de tiempo de recorrido.

Es un acercamiento mesoscópico descriptivo que extrapola a las vías en general, mediante procedimientos probabilísticos, los resultados obtenidos de análisis microscópicos basados en observaciones a lo largo del tiempo y en distintos lugares.

## Capítulo 3

# Descripción del modelo

### 3.1. La representación

#### 3.1.1. Datos de entrada

La *demanda* concentra la información de los viajes generados de acuerdo a las características de éstos y de los usuarios que los llevan acabo, a saber, el origen, destino y horario de partida de los viajes, así como los patrones de preferencia del conductor que los realiza. Clasificamos a los conductores según su tipo  $\tau$ , y llamamos  $\mathfrak{J}$  al *conjunto de tipos de conductores* o usuarios, de forma que  $\tau \in \mathfrak{J}$ . De esta manera la demanda se expresa en función de  $\tau$ .

El tipo  $\tau$  de *usuario* incluye el *origen*  $x$ , *destino*  $y$ , tipo de *vehículo*  $\gamma$  y las preferencias de usuario  $\omega$ . En este contexto definimos el tipo de usuario como  $\tau = (x, y, \gamma, \omega)$ . Decimos que dos usuarios son del mismo tipo si coinciden en origen, destino, tipo de vehículo y preferencias. Consideramos que cada viaje se realiza por usuarios distintos, lo cual es falso en la práctica; la desventaja de esta simplificación es no considerar viajes dependientes. Por ejemplo, el padre que deja a sus hijos en la escuela y luego va a su trabajo se modela con dos usuarios, uno que sale de su casa hacia la escuela a las 7:30 (para llegar a las 8:00) y otro que sale de la escuela hacia el trabajo a las 8:10 (suponiendo que llegó a tiempo a la escuela a las 8:00 y tardó 10 min en regresar al auto). Si el padre no llega a tiempo a la escuela tampoco saldrá a tiempo hacia el trabajo, pero para el modelo los viajes son independientes y, a pesar de que el primer usuario llegue a la escuela a las 8:20, el segundo partirá hacia su trabajo a las 8:10.

Llamamos  $V_x$  al *conjunto de orígenes*,  $V_y$  al *conjunto de destinos*,  $\Gamma$  al *conjunto de tipos de vehículo* y  $\Omega$  al *conjunto de preferencias de usuario*. Por cada preferencia de usuario, digamos

$p_i \in \Omega$ , asignamos al usuario  $\tau$  un valor  $\omega_i$  que corresponde a la  $i$ -ésima coordenada del vector  $\omega$ . Así agrupamos todas las preferencias del usuario  $\tau$  en un vector  $\omega_\tau$  y llamamos  $\bar{\Omega}$  al conjunto de éstos vectores. Entonces  $\omega_\tau = (\omega_1, \omega_2, \dots) \in \bar{\Omega} \subset \mathbb{R}^{|\Omega|}$ . De acuerdo a esto:

$$\mathfrak{J} \subseteq V_x \times V_y \times \Gamma \times \bar{\Omega}$$

La representación de la demanda consiste de la familia de *matrices de demanda*  $\{D_{ijk}\}_{(i, \gamma_j, \omega_k) \in \Lambda \times \Gamma \times \bar{\Omega}}$  donde  $\Lambda$  es el *lapso* de tiempo en que transcurre la jornada. Los renglones y columnas de cada una de las matrices  $D$  están dados por los orígenes y destinos de los usuarios. La familia consiste de una matriz  $D$  por cada terna de horario de salida, tipo de vehículo y preferencias de usuario. Por lo tanto, las entradas  $d_{ij}$  de cada matriz corresponden a la cantidad de vehículos del origen correspondiente a su fila y destino correspondiente a su columna, que comienzan su viaje con el horario, vehículo y preferencias correspondientes a la matriz.

Ya que nuestro interés se centra en el tránsito periódico o cotidiano, es necesario definir a qué cotidianidad o periodo nos referimos. Eso significa especificar el periodo, el lapso y los eventos que consideramos cotidianos. Definimos el *lapso*  $\Lambda$ , con mayor precisión, como el intervalo discretizado de tiempo que queremos modelar. Consiste del conjunto de tiempos en los que transcurrirá la jornada. El *periodo* es el tiempo en el que se repite el lapso. Por ejemplo, si escogemos los viernes de quincena, el periodo se fija en 15 días; el lapso, sin embargo, varía dependiendo de los eventos a considerar. Para tomar en cuenta la salida al trabajo por la mañana, las actividades de recreación características de esos días, y el regreso a la casa, incluso a altas horas de la madrugada, podemos escoger  $\Lambda = \{0, 1, 2, 3, \dots, 17280\}$ , cada 5 segundos desde las 6 am del viernes a las 6 am del sábado.

No estamos considerando el periodo como una variable explícita del modelo, sin embargo se encuentra implícita en la información de la demanda al igual que los eventos que se consideran cotidianos respecto al periodo y lapso establecidos. Los datos de la demanda deben de proveer la información necesaria para aquello que se desea modelar, si se trata de los martes por la mañana los datos deben corresponder a ese lapso, periodo y eventos precisos. Por exclusión, los viajes que no se consideren dentro de la demanda son para el modelo *viajes aleatorios*, no periódicos.

Es un modelo discreto en el tiempo. La unidad de tiempo que utilizamos es el paso  $\pi$ , que dura  $\pi s$  ( $\pi$  segundos, aunque omitimos los segundos en adelante) y se determina según los objetivos del modelo; como unidad prototipo utilizamos  $\pi = 5$ . Un tiempo  $t$  del modelo equivale a  $t \times \pi s$ . En este contexto cada paso del modelo corresponde a avanzar de  $t$  a  $t + 1$ , exactamente  $\pi s$ . El formato de la demanda, respecto al horario de salida de los viajes, debe incluir a  $\pi$  como la unidad de tiempo.

La *red vial* se representa como una gráfica dirigida  $G$  con valores paramétricos asociados a sus flechas y vértices; es decir, tomamos en cuenta los *cruceros* de la red vial como los *vértices* de nuestra gráfica, que denotamos  $V(G)$ , y los *segmentos básicos* como las *flechas* dirigidas de  $G$ , que denominamos  $F(G)$ . Un *segmento básico* es una parte de una avenida o calle, de un sólo

sentido y que no contiene cruceros, excepto por aquellos que la delimitan en sus extremos. Es decir, las cuerdas con doble sentido de circulación se toman como dos segmentos básicos, uno en un sentido y otro en el otro. Así cada segmento básico une (exactamente) dos crucesos con un cierto sentido y se representa, en nuestra gráfica, por una flecha que une (exactamente) los vértices asociados a esos dos crucesos y apuntando en el sentido que corresponde a la circulación. La figura 3.1 ejemplifica la representación de una pequeña red: La parte (a) muestra la red inicial con los sentidos de circulación, la (b) localiza los vértices y flechas que representarán dicha red, la (c) es la gráfica que el modelo utiliza como representación de la red inicial.

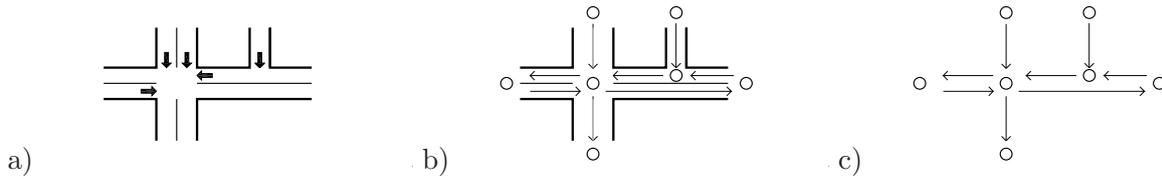


Figura 3.1: Representación de un cruceo como una digráfica.

Las características de la red consisten en su estructura, que se considera en la digráfica  $G$ , y las distintas características de las vías y crucesos que la conforman, los cuales se incorporan al modelo como características de flechas y vértices. Añadimos  $\mathcal{T}$ , el conjunto de tipos de vía de la red, el cual determina la subdivisión de las distintas arterias por el tipo de servicio para el que están diseñadas.

Las características de flechas son:<sup>1</sup>

- Longitud,  $\lambda : F(G) \rightarrow \mathbb{R}^+$   
Asigna a cada flecha  $a$  la longitud  $\lambda(a)$  del segmento que representa.
- Número de carriles,  $\kappa : F(G) \rightarrow \mathbb{N}$   
Asigna a cada flecha  $a$  el número de carriles  $\kappa(a)$  del segmento que representa.
- Carriles por operación,  $\kappa : F(G) \times F(G) \rightarrow \mathbb{R}^+$

Dado el diseño de la flecha  $a$ ,  $\kappa(a, b)$  es la proporción de la capacidad de la flecha  $a$  (definida en 2.3.2) que está proyectada para ir hacia la flecha  $b$ .

En el extremo de la flecha  $a$  los vehículos que esperan cruzar el vértice se obstruyen unos a otros mientras esperan cruzar. Estos vehículos se distribuyen, de acuerdo a la maniobra a realizar, en los diferentes carriles o zonas del extremo de la flecha, de modo que hay maniobras que se obstruyen mutuamente y otras que son independientes (vuelta izquierda y derecha suelen ser independientes en vías de varios carriles pero continuar recto suele obstruirse con ambas). Entonces, de la capacidad total de la flecha  $a$  consideramos, para cada maniobra, la proporción  $\kappa(a, b)$  de vehículos que pueden realizar la maniobra en cuestión.

<sup>1</sup>Utilizamos como unidades prototipo de medida el metro y el segundo.

Esta medida es independiente de las preferencias de paso  $\phi$  que considera el diseño del vértice y las preferencias del reglamento y señalización, mientras que  $\kappa(a, b)$  considera las obstrucciones de los vehículos dentro de la flecha  $a$  dado el diseño de ésta.

La función  $\kappa(a, b)$  se puede obtener a partir de diferentes criterios dado el diseño de la flecha siempre que respete la definición cualitativa que presentamos. Dado que es una proporción cualquier definición debe cumplir que:

$$\sum_{b \in N^+(v)} \kappa(a, b) = 1$$

donde  $v$  es el vértice final de  $a$ .

- **Velocidad de proyecto**,  $v : F(G) \rightarrow \mathbb{R}^+$   
Asigna a cada flecha  $a$  la velocidad  $v(a)$  para la que fue diseñada (proyectada).
- **Tiempo máximo de recorrido continuo**,  $\varpi : F(G) \rightarrow \mathbb{R}^+$   
Asigna a cada flecha  $a$  el tiempo máximo necesario  $\varpi(a)$  para recorrerla sin detenerse totalmente. Establece implícitamente, una velocidad a partir de la cual se considera que el vehículo está detenido.
- **Ancho de carril**,  $ac : F(G) \rightarrow [0, 3.60]$   
Medido en metros. El ancho de 3.60 es considerado ideal puesto que por arriba de este valor no se perciben mejoras en el funcionamiento<sup>2</sup>.
- **Alineación vertical**,  $av : F(G) \rightarrow [0, 1]$   
Se refiere a la pendiente y se calcula como  $\text{sen}(\beta)$  donde  $\beta$  es el ángulo (agudo) de la pendiente.
- **Condición de superficie de rodamiento**,  $csr : F(G) \rightarrow [-1, 1]$   
Vías cuya superficie hacen muy difícil transitar por ellas corresponden al valor -1. Aquellas en excelentes condiciones, por arriba del promedio (el promedio se representa con el valor 0) corresponden al 1.
- **Entorno urbano**,  $eu : F(G) \rightarrow [-1, 1]$   
Una vía que el usuario trata siempre de evitar por su entorno corresponde al parámetro -1. Una vía por la que se desea pasar a causa de su entorno corresponde al 1.
- **Tipo de vía**,  $\vartheta : F(G) \rightarrow \mathcal{T}$   
Tipo de vía,  $\vartheta(a)$ , correspondiente a la flecha  $a$ .

Con esta abstracción de las cuadras y esquinas de la ciudad nos es posible describir cómo están dispuestas las calles, avenidas y sus conexiones, las vueltas prohibidas, atajos escondidos, cruces complicados o problemáticos, pasos a desnivel, cuellos de botella, libramientos, etc.

---

<sup>2</sup>[MCC] página 81

Definimos  $N^+(v)$ , la *exvecindad* de  $v$ , como el conjunto de flechas que salen del vértice  $v$ . Análogamente  $N^-(v)$ , la *invecindad*, es el conjunto de flechas que entran al vértice  $v$ :

$$N^+(v) = \{a \in F(G) \mid a = (v, u), u \in V(G)\}$$

$$N^-(v) = \{a \in F(G) \mid a = (u, v), u \in V(G)\}$$

Consideramos solamente una característica de vértices, a saber:

- **Preferencia de paso**,  $\phi_v: N^-(v) \times N^+(v) \rightarrow [0, 1]$  Para cada arista  $a$  que llega a un vértice dado, asigna a cada una de las aristas  $b$  que salen de él la preferencia de paso,  $\phi_v(a, b)$ , por unidad de tiempo; es decir, la proporción del tiempo total en que un vehículo, que llega a un vértice por la flecha  $a$  y sale por la  $b$ , tiene preferencia de paso. Esto da cuenta de las preferencias marcadas por las diferentes señales (fijas o semáforos) y las de tipo de vía.

No estamos considerando el corte periódico a la circulación que producen los semáforos pero sí su efecto en las preferencias de paso. En el modelo, si en una intersección el flujo en cada ciclo es menor al que el semáforo es capaz de dejar pasar en dicho ciclo, los vehículos pasarán sin interrupción. En la práctica algunos vehículos que no llegaron en luz verde tendrían que esperar por ésta. El modelo produce colas en el caso contrario, en el que el flujo que llega en un ciclo de fases no logra desahogarse dentro del mismo. El caso que consideramos es un caso ideal en el sentido de sincronización semafórica. ([MCC] pág. 12)

El diseño de la gráfica y la selección y combinación de los parámetros nos permiten considerar e introducir en el modelo las características de infraestructura. De la misma forma, las características del sistema de actividades se incorporan por medio de la demanda, las preferencias de usuario y las características de su vehículo<sup>3</sup>. Denotamos  $\gamma_\tau$  al vehículo del usuario de tipo  $\tau$ , sus características son:

- **Tamaño de vehículo**,  $\varsigma: \Gamma \rightarrow \mathbb{R}$   
Coeficiente que determina el espacio,  $\varsigma(\gamma)$ , que ocupa un vehículo de tipo  $\gamma$  dentro de una cuadra, según su tamaño.
- **Conjunto de vías aptas**,  $\theta: \Gamma \rightarrow 2^\Gamma$   
Conjunto de los tipos de vías,  $\theta(\gamma)$ , a los que el vehículo de tipo  $\gamma$  tiene acceso.

$$\theta(\gamma) \subseteq \Gamma$$

<sup>3</sup>Las preferencias, origen-destino y características de vehículo conforman las características de usuario.

- Conjunto *adicional* de vías,  $\theta^* : \Gamma \rightarrow 2^{F(G)}$   
Conjunto *adicional* de vías,  $\theta^*(\gamma)$ , a las que el vehículo de tipo  $\gamma$  tiene acceso (por ejemplo, rutas del sistema de transporte de pasajeros).

$$\theta^*(\gamma) \subseteq F(G)$$

Estas características nos permiten considerar las diferentes relaciones que los distintos vehículos guardan conforme a velocidad, condiciones requeridas y recursos de red utilizados.

Un tiempo de viaje corto no es la única cualidad que buscan los usuarios al elegir sus rutas. Para tomar en cuenta las distintas cualidades de las rutas incluimos el *conjunto de preferencias de usuario*, que reúne todas las preferencias excepto la referente a vías ideales y denotamos como  $Pref^4$ :

- Preferencia por vías principales,  $pvp \in \mathbb{R}^+$ .
- Preferencia por vías con buenas condiciones de la superficie de rodamiento,  $psr \in \mathbb{R}^+$ .
- Preferencia por entorno urbano,  $peu \in \mathbb{R}^+$ .  
En términos de seguridad, panorama, salubridad, etc.
- Preferencia por vías ideales,  $P_I \in \mathbb{R}^+$ .  
Preferencia por una vía que posee todas las características favorables.

De este modo representamos la preferencia por caminos más seguros, más agradables y más comunes. Las unidades de estas magnitudes se dan en la sección 3.6 de justificación.

Es un modelo *semi-agregado*. Modelamos el comportamiento de grupos de usuarios (conductores), no de cada usuario ni del agregado de todos los usuarios. Agrupamos los usuarios de acuerdo a las características que consideramos de modo que sólo estén en un mismo grupo usuarios indistintos para el modelo. Esto nos da libertad en el sentido de que, como grupo, podemos suponer un comportamiento vehicular más homogéneo, tratando las características no explícitas del modelo de forma probabilística pero sin perder aquellas que creemos fundamentales para la dinámica del tránsito en la red.

Los vehículos dentro del modelo constituyen el flujo en las flechas de la gráfica. De hecho, los *actores* del modelo son los vértices y las flechas. En este sentido definimos el *flujo general*,  $F(a, t)$ , de una flecha  $a$  en el tiempo  $t$  como la cantidad total de vehículos que se encuentran en dicha flecha en el tiempo  $t$ . Entonces, el *patrón de flujos* consiste en el desarrollo del flujo a lo largo del lapso, es decir, de los valores de  $F(a, t) \forall a \in F(G), t \in \Lambda$ .

---

<sup>4</sup>Ver 3.6.2

### 3.1.2. Capas de usuario

Para dar seguimiento a cada tipo de usuario descomponemos el flujo distinguiendo los diferentes usuarios que lo conforman. Lo anterior nos lleva a definir el *flujo parcial*,  $F_\tau(a, t)$ , como la cantidad de usuarios de tipo  $\tau$ , que se encuentran sobre la flecha  $a$  en el tiempo  $t$ .

El seguimiento de los flujos parciales se lleva a cabo en copias parciales de la red que llamamos *capas*. De esta forma podemos considerar por separado el flujo parcial de cada tipo de usuario  $\tau$  y el *flujo general*, integración de los parciales, de donde se desprende la siguiente relación.

$$F(a, t) = \sum_{\tau \in \mathcal{T}} F_\tau(a, t) \quad (3.1)$$

La porción de la red vial que utiliza cada tipo de usuario es muy pequeña con respecto al tamaño de la red completa. El tipo  $\gamma_\tau$  de vehículo restringe las posibles vías a utilizar (por ejemplo las restricciones de paso a vehículos de carga); el origen-destino determina una zona útil de la red si consideramos rutas cuya longitud no exceda cierta longitud máxima. Por lo tanto es posible reducir la red, para cada usuario, quitando las *vías no aptas* para su *tipo de vehículo* y todo aquello que no sea asequible por una ruta cuya longitud no exceda un valor máximo y que una el origen y destino del usuario  $\tau$  del vehículo  $\gamma_\tau$  en cuestión.

Reducimos la red para cada tipo  $\tau$  de usuario de la siguiente manera: consideramos el tipo de vehículo  $\gamma$  del usuario  $\tau$  y, a partir de la digráfica  $G$ , construimos la *sub-digráfica de vías aptas*  $G_\gamma$  que consiste de las vías aptas  $\theta(\gamma)$  y el conjunto adicional de vías aptas  $\theta^*(\gamma)$  de  $\gamma$ . De ahí construimos la *subdigráfica acotada*  $G_\tau$  como la subdigráfica de  $G_\gamma$  formada por los vértices y flechas por los que pasa una ruta de  $x$  hacia  $y$  (una  $xy$ -ruta), cuya longitud no exceda la de la  $xy$ -ruta más corta en más de cierto número  $\sigma \geq 0$ , que llamamos *radio de vecindad*. Para establecer la longitud de las rutas, consideramos la *distancia a flujo libre desde el origen  $x$  y hacia el destino  $y$* ,  $d_x^r(v)$  y  $d_y^r(v)$ , que definimos como el tiempo de recorrido mínimo, cuando la red está vacía, entre  $x$  y  $v$  o  $v$  y  $y$  respectivamente<sup>5</sup>.

Por lo tanto definimos:

- El *conjunto general*  $\Theta_\gamma$  de *vías aptas* para el tipo  $\gamma \in \Gamma$  como:

$$\Theta_\gamma = \{a \in F(G) \mid \vartheta(a) \in \theta(\gamma)\} \cup \{a \in F(G) \mid a \in \theta^*(\gamma)\}$$

<sup>5</sup>Va más allá de los fines de esta sección una definición formal de estas distancias y de las rutas, como caminos en el tiempo; sin embargo, la definición formal de estos conceptos está en el apéndice A.1.2 para distancias a flujo libre y A.2 para rutas.

Es decir, el conjunto de todas las flechas  $a$  que cumplan que: el tipo de vía de  $a$  es apto para el tipo de vehículo  $\gamma$  ó  $a$  está en el conjunto de vías aptas adicionales de  $\gamma$ .

- $G_\gamma = G[\Theta_\gamma]$ . Es decir, la subdigráfica que se obtiene de  $G$  tomando solamente las flechas de  $\Theta_\gamma$  y los vértices de sus extremos.
- La *vecindad*  $H_\tau$  como el conjunto de los vértices aseguibles para el tipo de usuario  $\tau$ , en el siguiente sentido:

$$H_\tau = \{v \in V(G_\gamma) \mid d_x^\tau(v) + d_y^\tau(v) < d_x^\tau(y) + \sigma\}$$

- Definimos la *sub-digráfica acotada* como  $G_\tau = G_\gamma[H_\tau]$ . Es decir, la subgráfica que se obtiene de  $G_\gamma$  tomando solamente los vértices de  $H_\tau$  y las flechas que estén entre ellos.

Para llevar a cabo la *inyección de flujo* (ver 3.3.3) es necesaria una *flecha auxiliar*  $h_\tau$ . Esta flecha no requiere de vértice inicial, su vértice final es el origen  $x(\tau)$ . El conjunto  $\Xi = \{h_\tau \mid \tau \in \mathbb{I}\}$  se define como el *conjunto de flechas auxiliares*. La digráfica  $G_\tau^*$  constituye la *capa* del tipo de usuario  $\tau$  y se obtiene de extender la digráfica  $G_\tau$  adhiriendo la flecha  $h_\tau$  a su conjunto de flechas, es decir:

$$F(G_\tau^*) = F(G_\tau) \cup h_\tau \quad y \quad V(G_\tau^*) = V(G_\tau).$$

De la misma forma la *digráfica extendida*  $G^*$  es la extensión de  $G$  cuando se adhiere el conjunto  $\Xi$  a su conjunto de flechas, es decir:

$$F(G^*) = F(G) \cup \Xi \quad y \quad V(G^*) = V(G).$$

## 3.2. El problema

El problema que nos planteamos consiste en determinar el patrón de flujos del tránsito periódico vehicular urbano, dada la red vial y sus características, los tipos y características de los usuarios y sus vehículos, y la información de la demanda; es decir, los volúmenes de vehículos, de cada tipo, que parten de un origen y hacia un destino fijos, en un horario establecido; tomando en cuenta todos los orígenes, destinos y horarios. Se trata de modelar la primera relación principal (ver 1.1) que consiste en determinar el patrón de flujos a partir de la infraestructura de transporte y el sistema de actividades. Lo que se traduce en encontrar, para el tránsito periódico, los valores de  $F(a, t) \forall a \in F(G), t \in \Lambda$  a partir de  $\{D_{ijk}\}_{(i, \gamma_j, \omega_k) \in \Lambda \times \Gamma \times \bar{\Omega}}$ ,  $G$  y de las funciones características de flechas, vértices, vehículos y preferencias de usuarios.

### 3.2.1. Heurísticamente

Nuestra forma de resolverlo es recreando el proceso mediante el cual los usuarios construyen sus costumbres de transporte, en una dinámica de prueba y error a partir de la red, sus necesidades, la interacción con otros usuarios y las experiencias que van acumulando en el proceso. Planteamos que es posible encontrar el patrón de flujos periódico (o cotidiano) de una cierta manera constructiva que describamos a continuación:

Imaginemos una serie de días totalmente *comunes*, o quizá no tanto en el sentido de que son días en que no pasa *nada extraordinario*, no hay choques que obstruyan avenidas, ni lluvias, ni ÉL CLÁSICO en el estadio olímpico; es un día en que en cualquier punto de la ciudad no pasa nada más de lo que pasa todos los días. Ahora imaginemos que todas las personas de la ciudad olvidaron, no lo que tienen que hacer, o dónde, sino las rutas que acostumbran recorrer para llegar a esos lugares.

A pesar del olvido todos conocen tanto sus *horarios y actividades* como la ciudad misma, así que no se detiene el movimiento urbano. Cada *vehículo* sale de su *origen* en su horario habitual y al llegar a cada bifurcación escoge el camino que más le convenga de acuerdo a la *situación*, es decir al tiempo que el *conductor* estima le tomará llegar a su destino dada la *congestión*, las *características físicas*, la *ubicación* de cada vía y a su *experiencia*, que por el momento es nula. Conforme avanza se encuentra con otros vehículos que ocupan su misma vía y que van en la misma dirección que él, quizá con su mismo origen o quizás no, van y vienen de cualquier otra parte y sólo comparten con él esa calle. El vehículo avanza normalmente como las condiciones de las calles y la presencia de los otros vehículos se lo permiten. Al llegar a su *destino* hace *memoria* de lo sucedido durante su viaje y se informa hasta enterarse de cómo se desarrolló la jornada en sus posibles rutas, es decir, de cuánto tiempo se tardaron los vehículos en recorrer cada calle dependiendo de la hora.

Al otro día, también perfectamente normal, vuelve a salir de su punto de origen con el mismo horario, pero al elegir su camino toma en cuenta tanto la *situación* del momento como la *experiencia* que recopiló de la jornada anterior. Ya en su destino, vuelve a recopilar la información del tránsito reemplazando la del día anterior. Pasan los días hasta que encuentra que él y cada uno de sus compañeros realizan sus viajes exactamente de la misma forma cada día, o quizá no exactamente de la misma forma, pero sí que se parecen mucho todos los días o que varían mucho de día a día pero son finalmente las mismas 4 ó 5 rutas de siempre. Entonces reconoce sus costumbres de transporte.

### 3.2.2. Formalmente

Comenzamos con  $F(a, 0)_1, \forall a \in F(G)$ . A partir de una serie de procesos encontramos  $F(a, 1)_1 \forall a \in F(G)$  y siguiendo un proceso inductivo terminamos con  $F(a, t)_1 \forall a \in F(G), t \in \Lambda$ . Al patrón de flujos resultado de este proceso inductivo sobre  $t$  lo llamamos *jornada* y denotamos

$J_n = \{\{F(a, t)_n\}_{a \in F(G)}\}_{t \in \Lambda}$ . Es al número de jornada a lo que se refiere el subíndice  $n$ , lo mismo que el del flujo.

Completada la primera jornada  $J_1$  comenzamos la segunda con el mismo valor  $F(a, 0)_1 = F(a, 0)_2$  introduciendo la experiencia de  $J_1$  mediante lo que definiremos como *memoria de retroalimentación* (la cual denotaremos por  $MR_n$  y describimos con precisión en la sección 3.3.2). Completamos  $J_2$  y después de ella  $J_3$ , en un segundo proceso de inducción que finalmente evaluaremos para determinar si converge o no a una jornada límite.

En resumen el problema es, dada una *función de iteración*  $F: \mathbb{N}^{F(G)} \rightarrow \mathbb{N}^{F(G)}$  tal que:

$$F(\{F(a, t)_n\}_{a \in F(G)}) = \{F(a, t+1)_n\}_{a \in F(G)}$$

y una *super-función de iteración*  $\mathfrak{S}: (\mathbb{N}^{F(G)})^\Lambda \rightarrow (\mathbb{N}^{F(G)})^\Lambda$  tal que:

$$\mathfrak{S}(\{\{F(a, t)_n\}_{a \in F(G)}\}_{t \in \Lambda}) = \{\{F(a, t)_{n+1}\}_{a \in F(G)}\}_{t \in \Lambda}$$

o, tomando  $J_n = \{\{F(a, t)_n\}_{a \in F(G)}\}_{t \in \Lambda}$ :

$$\mathfrak{S}(J_n) = J_{n+1}$$

encontrar  $\{\{F(a, t)_m\}_{a \in F(G)}\}_{t \in \Lambda} = J_m$  tal que:

$$\mathfrak{S}(J_m) = J_m$$

donde  $F$  actúa sobre cada tiempo contemplando las reglas por las que se desarrolla el tránsito en un día normal, incluyendo la demanda, la gráfica y sus características, y los usuarios; y  $\mathfrak{S}$  actúa sobre las jornadas considerando la repercusión en el patrón de flujos de la experiencia de los conductores.

### 3.3. Descripción de una jornada

El proceso por el que construimos una jornada consta de un ciclo, de dos etapas, que se repite en cada *paso* (en el que se avanza del tiempo  $t$  al  $t+1$ ) hasta llegar al final del *lapso*  $\Lambda$ . Cada uno de estos ciclos hace las veces de (lo que formalmente es) la regla de iteración  $F$ .

En cualquier momento, a lo largo de la jornada, los vehículos que conforman el flujo se encuentran dentro de una flecha, puesto que los que aún no salen de su origen y los que llegaron a su destino no se consideran parte del flujo. Distinguimos dos estados de un vehículo dentro de una flecha, cuando avanza sin detenerse y cuando está detenido por una cola. A estos dos estados corresponden las dos etapas que conforman un *paso*, (1) *el vehículo se encuentra avanzando sin detenerse*, al llegar al final de la flecha (2) *se forma en la cola para cruzar el vértice*, elige su siguiente flecha con el propósito de alcanzar su destino y la toma; regresando a la etapa (1) en la nueva

flecha. La presente sección expone el monitoreo de estas etapas y las transiciones entre una y otra, así como la justificación precisa de estas distinciones.

Los procesos que describimos a continuación se basan en consideraciones particulares de fenómenos como la congestión, la demora por cola, demora por fricción, selección de vías, y en general del fenómeno del tránsito. Algunas de estas consideraciones están fijas en la estructura del modelo (como el supuesto de que el fenómeno del tránsito se puede simular de forma continua en el tiempo), mientras que otras se encuentran agrupadas en las definiciones prototipo de las funciones especiales que se definen a lo largo de esta sección. Dichas funciones están pensadas para poder ser modificadas por el implementador del modelo, de modo que se ajusten al caso particular que éste requiera. Las definiciones prototipo que presentamos están inspiradas en los acercamientos brevemente descritos en el capítulo 2, junto con consideraciones nuestras. De cualquier forma, la definición de cada una de las funciones especiales se justifica y expone a detalle en la subsección que lleva su nombre, dentro de la sección 3.6.

### 3.3.1. Seguimiento de vehículos

Del recorrido de un vehículo a través de una flecha, distinguimos la parte en la que estuvo avanzando de aquella en la que estuvo detenido por una cola. Dentro del modelo, los vehículos avanzan continuamente hasta llegar al extremo, donde se detienen en la *cola* hasta cruzar el vértice; es decir, juntamos los momentos estáticos en una etapa de *cola*, al final del recorrido, y los momentos de movimiento en una etapa de *recorrido continuo*.

Para definir claramente el estado estático y el estado de movimiento, utilizamos el parámetro  $\varpi(a)$  que se define como el *tiempo máximo de recorrido continuo* de la flecha  $a$ , sin que el vehículo se detenga en una cola. Visto de otra forma, este tiempo máximo corresponde a una velocidad mínima (a saber,  $\frac{\lambda(a)}{\varpi(a)}$ ) en la que consideramos que un vehículo se encuentra en movimiento continuo (quizá muy lento, forzado, pero continuo); por debajo de dicha velocidad el modelo entiende que el vehículo está en una cola y su movimiento responde a los avances parciales típicos de esta.

Los flujos  $F(a, t)$  y  $F_\tau(a, t)$  dan cuenta de los vehículos que se encuentran recorriendo la flecha  $a$ , sin embargo no distinguen si un vehículo está en la etapa de recorrido continuo o de cola. Para dar seguimiento a la etapa de movimiento continuo y determinar el momento en que los vehículos llegan al extremo de la flecha y entran a la siguiente etapa, creamos el *monitoreo de flecha*  $M_\tau(a, t)$  que nos indica el avance de los grupos de vehículos de tipo  $\tau \in \mathfrak{J}$  en el recorrido de la flecha  $a$ , en el tiempo  $t$ .

Definimos  $M_\tau(a, t)$  como el conjunto de *grupos de vehículos en recorrido continuo*  $(m, \delta)$ , donde para cada pareja del conjunto  $m$  es el número de vehículos de tipo  $\tau$  a los que les hace falta  $\delta$  unidades de tiempo para cruzar la flecha  $a$ . En un principio se establece un valor inicial para  $\delta$ ,

conforme avanza el tiempo  $t$  el valor de  $\delta$  disminuye hasta  $\delta = 0$  que indica que los vehículos del tipo  $\tau$  terminaron el recorrido continuo y se dice que llegaron al vértice final de la flecha.

Llamamos  $\mu_\tau(a, t)$  al número de *vehículos monitoreados de tipo*  $\tau$ , y  $\mu(a, t)$  al total de *vehículos monitoreados*, de todos los tipos  $\tau \in \mathfrak{J}$  que se encuentran en alguna  $M_\tau(a, t)$ . Por lo tanto se cumple:

$$\mu(a, t) = \sum_{\tau \in \mathfrak{J}} \mu_\tau(a, t).$$

A partir de esta magnitud calculamos  $\lambda_M(a, t)$ , el *espacio ocupado en recorrido continuo* de la flecha  $a$  en el tiempo  $t$ , como:

$$\lambda_M(a, t) = \varsigma(\gamma_\tau) \times \mu_\tau(a, t)$$

utilizando la función  $\varsigma$  que nos indica el tamaño del vehículo  $\gamma_\tau$  del usuario  $\tau$ .

Al extremo de toda flecha, en el vértice final, hay una cola de vehículos para cruzar dicho vértice. Cuando un grupo de vehículos termina el recorrido de la flecha (i.e. cuando  $\delta = 0$ ) se coloca al final de dicha cola, terminando la etapa de movimiento continuo y comenzando la de cola. La cola avanza conforme a la *congestión, preferencias de paso y espacio libre* en la exvecindad  $N^+(v)$  del vértice  $v$ .

El *monitoreo de cola*  $\zeta(a, t)$ , es el encargado de dar seguimiento a este proceso. Se define como el conjunto de *grupos de vehículos en cola*  $(m, b, \tau)$ . Para cada grupo de vehículos,  $m$  es la cantidad de vehículos del grupo,  $\tau \in \mathfrak{J}$  es el tipo de vehículo del grupo (todos los vehículos de un grupo son del mismo tipo) y  $b \in N^+(v)$  la flecha destino<sup>6</sup>. Cuando un vehículo (o grupo de) se incorpora a la cola, su flecha destino  $b$  se establece como  $b = \emptyset$ . Cuando el vehículo en cuestión está suficientemente adelante en la cola (de manera que le es posible cruzar el vértice en el siguiente *paso*) su flecha destino es seleccionada con base en los criterios que se exponen en el siguiente apartado (ver 3.3.2).

Definimos  $\Upsilon(a, t)$ , el número total de *vehículos en cola* como:

$$\Upsilon(a, t) = \sum_{(m, b, \tau) \in \zeta(a, t)} m.$$

Llamamos  $\lambda_\zeta(a, t)$  al *espacio ocupado en cola* dentro de la flecha  $a$ , el cual se calcula con la siguiente fórmula:

$$\lambda_\zeta(a, t) = \sum_{(m, b, \tau) \in \zeta(a, t)} \varsigma(\gamma_\tau) \times m.$$

Existe una sencilla relación entre estas funciones de seguimiento en las aristas y el flujo dentro de ellas, misma que expresamos como sigue:

$$F(a, t) = \mu(a, t) + \Upsilon(a, t). \quad (3.2)$$

---

<sup>6</sup>El vértice  $v$  es el que se encuentra al final de la flecha  $a$ , de modo que es también el inicial de la flecha  $b$ , dado que  $b \in N^+(v)$ .

Denotamos  $\lambda_\emptyset$  al *espacio libre* en la flecha  $a$  al tiempo  $t$ , el cual se define como la longitud total de los carriles de la flecha  $a$  menos la longitud ocupada por los vehículos que se encuentran en ella. Se calcula como:

$$\lambda_\emptyset(a, t) = \lambda(a) \cdot \kappa(a) - \lambda_\zeta(a, t) - \lambda_M(b, t).$$

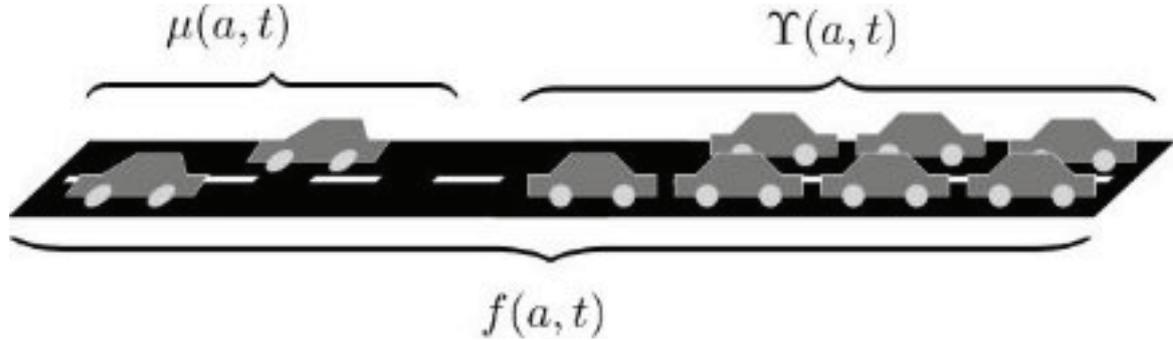


Figura 3.2: Flujo en recorrido continuo y cola.

FIGURA 3.2 ( $\mu$  y  $\Upsilon$ , R)

Al tiempo total en movimiento de un vehículo, en el recorrido de la flecha  $a$ , lo llamamos *tiempo de recorrido continuo* de  $a$ . En el modelo lo predecimos, en cada tiempo  $t$ , con la función  $TRC(a, t)$  que considera las características de la arista  $a$  y el flujo general del momento. Es esta función la que calcula, para el tiempo  $t$  en que comienza el recorrido, el valor inicial  $\delta$  del monitoreo  $M_\tau(a, t)$ , es decir:  $\delta = TRC(a, t)$ . Como función prototipo  $TRC(a, t)$  tenemos:

$$TRC(a, t) = \min \left\{ \varpi, \max \left\{ \frac{\lambda(a)}{v(a)}, \frac{\lambda(a)}{\left( -10 + \sqrt{100 + (20 \times 3.28) \frac{\lambda_\emptyset(a, t)}{\mu(a, t)}} \right) \times 1.609} \right\} \right\} \quad (3.3)$$

donde  $\lambda_\emptyset(b, t)$  es el espacio libre en la flecha  $b$  al tiempo  $t$ .

El tiempo que tarda un vehículo en una cola hasta llegar a su siguiente flecha se conoce como *demora por cola* y es de gran utilidad tanto para estudios de tránsito como para los procedimientos del modelo. Sin embargo no nos es posible medir esta magnitud puesto que no hacemos un seguimiento vehículo a vehículo. La función  $\chi(a, t)$  se encarga de estimar dicho valor utilizando la función  $\iota$ , que representa la cantidad de vehículos que cruzan el vértice en cada paso y se conoce como intensidad de paso<sup>7</sup>. Como prototipo para  $\chi$  proponemos:

$$\chi(a, t) = \begin{cases} \frac{\Upsilon(a, t-1)}{\iota(a, t-1)} & \text{si } \iota(a, t-1) > 0; \\ \infty & \text{si } \iota(a, t-1) = 0 \end{cases} \quad (3.4)$$

<sup>7</sup>Ver la definición en la pág. 31 de esta sección.

donde  $\overline{\infty}$  es un valor infinito que, además<sup>8</sup>, cumple  $1, 2, \dots < \overline{\infty} < \infty$ .

El tiempo total que tarda un vehículo en recorrer la flecha  $a$  es la suma del tiempo en recorrido continuo y el tiempo en cola. Llamamos a este valor *tiempo de recorrido*  $TR(a, t)$  y se define como:<sup>9</sup>

$$TR(a, t) = \begin{cases} TRC(a, t) + \chi(a, t) & \text{si } TR(a, \bar{t}) \leq TRC(a, t) + \chi(a, t) + (t - \bar{t}); \\ TR(a, \bar{t}) - (t - \bar{t}) & \text{si } TR(a, \bar{t}) > TRC(a, t) + \chi(a, t) + (t - \bar{t}) \end{cases} \quad (3.5)$$

donde  $\bar{t}$  es el último tiempo, anterior al actual  $t$ , en que el tiempo de recorrido fue finito.

Para el desahogo de la cola el modelo determina la cantidad de vehículos que saldrán de ésta y cruzarán el vértice final, digamos  $v$ , de la flecha en la que se encuentran. Para ésto consideramos el espacio libre en las flechas destino a las que se dirigen los vehículos, el diseño y la congestión del vértice por atravesar. Entendemos la congestión como el resultado de un conflicto por atravesar el mismo espacio, lo que representamos para el vértice  $v$  comparando el número de vehículos que pretenden atravesarlo y la operación que cada uno hará (de que flecha de entrada a qué flecha de salida pretende cruzar) con las capacidades de las flechas de salida para recibir vehículos (espacio libre).

Denotamos  $\rho(v, t) \in [0, 1]$  a la *congestión* del vértice  $v$  en el tiempo  $t$ . Como función prototipo tenemos:

$$\rho(v, t) = \frac{\sum_{b \in N^+(v)} \min\left(1, \frac{\lambda_\theta(b, t)}{D_\lambda(b, t)}\right)}{\|N^+(v)\|} \quad (3.6)$$

donde  $\lambda_\theta(b, t)$  es el espacio libre en la flecha  $b$  al tiempo  $t$  y  $D_\lambda(b, t)$  es la *demanda de espacio* en  $b$ , que se define como el espacio lineal (suma de longitudes) que ocupan los vehículos que están esperando cruzar el vértice  $v$  hacia la flecha  $b$  en el tiempo  $t$ . Consideramos  $\frac{\lambda_\theta(b, t)}{D_\lambda(b, t)} > 1$  en caso de indeterminación, cuando  $D_\lambda(b, t) = 0$ .

El diseño, señales y semáforos de un crucero dan distintas preferencias de paso de cada vía de entrada hacia cada una de las vías de salida (ver 1.1), lo que modelamos mediante las *preferencias de paso*  $\phi_v(a, b)$ . Por otro lado el flujo de una vía, al llegar al extremo, se distribuye en los carriles de ésta de acuerdo a su disposición, eligiendo el que facilite la operación a realizar de cada vehículo.

<sup>8</sup>Si consideramos los reales extendidos  $\mathbb{R}^* = \mathbb{R} \cup \infty$ , entonces  $\overline{\infty}$  cumple que  $\mathbb{R}^* \simeq \mathbb{R} \cup \overline{\infty}$  y  $\overline{\infty} < \infty$ . De este modo las operaciones de  $\overline{\infty}$  con cualquier otro  $r \in \mathbb{R}$  son aquellas de  $\mathbb{R}^*$  y no utilizamos operaciones entre  $\infty$  e  $\overline{\infty}$ .

Este valor simplemente nos sirve para distinguir el origen del valor infinito en  $TR$ : en el caso de  $\overline{\infty}$  se debe a la función  $\chi$  mientras que  $\infty$  proviene de las preferencias de usuario (ver 3.3.2). Definimos  $\overline{\infty} < \infty$  puesto que el valor de  $\chi$  puede volverse finito después de algunos pasos, lo que no pasa con el valor de  $1 + P_\tau$  que es constante en el tiempo. Esta consideración se utiliza en el algoritmo dinámico de Dijkstra a modo de saber si el valor de infinito de  $TR(a, t)$  implica que no debe considerarse la flecha en cuestión o si se trata de una obstrucción momentánea y la flecha se podrá utilizar en un tiempo posterior.

<sup>9</sup>La desigualdad en esta definición responde a la *condición de fila* del algoritmo dinámico de Dijkstra, ver A.2.1

Para tomar ésto en cuenta el modelo utiliza los *carriles por operación*  $\kappa(a, b)$ . Estos dos factores inciden en la *intensidad de paso*  $\iota(a, t)$  que definimos como el número de vehículos en la flecha  $a$  que pueden cruzar el vértice final de ésta (digamos  $v$ ), en el paso  $t$ . Dicha magnitud se obtiene mediante las preferencias de paso  $\phi(a, b)$ , los carriles por operación  $\kappa(a, b)$  y la congestión  $\rho(v, t)$  como sigue:

$$\iota(a, t) = \left\lfloor 0.5 \cdot \rho(v, t) \cdot \pi \cdot \kappa(a) \sum_{b \in N^+(v)} \phi(a, b) \cdot \kappa(a, b) \right\rfloor \quad (3.7)$$

donde  $v$  es el vértice final de  $a$ .

Por lo tanto, en cada paso  $\pi$  cruzarán el vértice  $\iota(a, t)$  vehículos de  $a$  (redondeado hacia abajo); siempre y cuando haya espacio en sus respectivas flechas destino.

En muchos de los casos el procedimiento de cola es mero protocolo, de hecho se dice que hay una cola sólo cuando, al final de cada paso,  $\zeta(a, t) \neq \emptyset$ , de modo que no hay tal si  $\zeta_\tau(a, t) = \emptyset \forall \tau \in \mathbb{J}$ . Es decir, consideramos una cola efectiva cuando hay al menos un vehículo que permaneció en ella, no cuando los vehículos solamente pasaron por ella simbólicamente, sin detenerse.

Consideramos que un vehículo está suficientemente adelante en una cola en el tiempo  $t$  si su número de lugar es menor al número de vehículos, de la flecha en la que se encuentra, que cruzarán el vértice en ese tiempo (es decir  $\iota(a, t)$ ). Cuando un vehículo está suficientemente adelante en la cola, evalúa la *efectividad* de las opciones de ruta que le presenta el vértice, escoge una y se dirige hacia ella. Así vuelve a entrar a otra flecha y recomienza el ciclo de seguimiento dentro de ella.

### 3.3.2. Criterios de selección de ruta

Lo principal para escoger una ruta es que lleve al vehículo al destino deseado sin dar vueltas innecesarias, el modelo resuelve ésto creando una *noción de ubicación*. La función  $d_y^\tau(v, t)$  provee de dicha noción etiquetando cada vértice  $v \in V(G_\tau)$  con una *distancia dinámica*<sup>10</sup> al vértice destino. Definimos dicha distancia como la duración del *recorrido más corto*, de  $v$  hacia  $y$ , que haya comenzado en el tiempo  $t$ .

La noción más común de distancia se basa en la longitud de la ruta más corta. La longitud es insensible a cambios en el flujo de una vía; no es diferente la longitud de una vía cuando está vacía o congestionada. Ya que la dinámica que nos interesa es la del flujo, escogimos como base de la distancia el *tiempo de recorrido* que tiene una estrecha relación con el flujo y la longitud de cada segmento básico. El tiempo de recorrido total de la misma ruta saliendo cinco minutos antes o

<sup>10</sup>Estamos abusando de la notación puesto que  $d_y^\tau$  está definida como la *distancia a flujo libre*, sin embargo deben poder distinguirse por el contexto ya que la distancia dinámica depende del tiempo de salida y el vértice mientras que la de flujo libre sólo del vértice.

después (por ejemplo, del inicio de la hora pico) puede variar enormemente, es por ésto que la función de distancia dinámica incorpora el *horario de salida* como argumento, dando una noción de ubicación respecto a la red, el flujo y sus horarios.

El tiempo de recorrido no es la única magnitud que distingue una buena ruta de una mala, consideramos también, la seguridad, lo agradable de la ruta, etc. Para unificar valores y tomar en cuenta las preferencias de usuario utilizamos la *preferencia global*  $P_\tau(a)$ , que resume las preferencias del usuario  $\tau$  respecto a las características de la flecha  $a$ ; entre más afines sean las características de la vía respecto a las preferencias de usuario, más grande será  $P_\tau(a)$ , con un valor máximo  $P_I$ . Consideramos entonces un conjunto de preferencias que denotamos<sup>11</sup>  $Pref = \{p_1, p_2, \dots, p_s\}$  que guarda una relación uno a uno con el conjunto  $Car = \{c_1, c_2, \dots, c_s\}$  de características de las flechas (a cada preferencia le corresponde una característica, es decir la preferencia y características de entorno urbano se corresponden), de modo que definimos  $P_\tau$  como:

$$P_\tau(a) = \begin{cases} \left( \sum_{1 \leq i \leq s} c_i \times p_i \right) & \text{si } \sum_{1 \leq i \leq s} c_i < 1; \\ \left( \sum_{1 \leq i \leq s} c_i \left( p_i + \frac{P_I - Q}{n-1} \right) \right) + \frac{Q - P_I}{n-1} & \text{si } \sum_{1 \leq i \leq s} c_i \geq 1 \end{cases} \quad (3.8)$$

donde  $Q = \sum_{1 \leq j \leq s} p_j$ .

El valor  $P_\tau(a)$  determina  $\widehat{TR}_\tau(a, t)$ , un nuevo *tiempo de recorrido, relativo* al usuario cuyas preferencias resume  $P_\tau$  y se define como:

$$\widehat{TR}_\tau(a, t) = \begin{cases} \frac{TR(a, t)}{1 + P_\tau(a)} & \text{si } 1 + P_\tau(a) > 0; \\ \infty & \text{si } 1 + P_\tau(a) \leq 0. \end{cases} \quad (3.9)$$

El *tiempo de recorrido relativo*  $\widehat{TR}_\tau$  induce una *distancia relativa* al usuario  $\tau$  que llamamos  $\widehat{d}_y^\tau(v, t)$  que es sensible a la *percepción* (entendida como sus preferencias respecto a ciertas características) de cada tipo de usuario. Cuando las características de una flecha son en suma inconvenientes para un usuario  $\tau$ , de forma que éste no esté dispuesto a tomar la flecha (si es demasiado insegura, por ejemplo) el tiempo de recorrido  $\widehat{TR}_\tau$  se vuelve infinito haciendo la flecha inaccesible (al menos para quien quiere llegar a su destino) y por lo tanto podemos no tomarla en cuenta en la capa relativa al usuario  $\tau$ . De esa forma la gráfica (capa) cambia su estructura topológica de acuerdo a las preferencias del usuario.

Las distancias son criterios *globales* en cuanto a la red, es decir, requieren de información global, de puntos lejanos a la posición del vehículo en cuestión. La información global proviene del conductor del vehículo: de su *conocimiento de la red* y de la *experiencia adquirida* cada vez que

<sup>11</sup> $s \in \mathbb{N}$  es tal que  $\|Car\| = \|Pref\| = s$

realiza el viaje. Es posible considerar como *viajes periódicos* aquellos realizados cotidianamente por usuarios distintos (pero del mismo tipo) como los trámites burocráticos. En ese caso la experiencia tiene menor influencia, aunque existe la transmitida entre usuarios; representamos esta situación con las *preferencias por vías principales*, de modo que los usuarios *no habituados* a un viaje tiendan a escoger el camino más fácil, el de las vías principales.

Las repeticiones del viaje las representamos haciendo *iteraciones* de la *jornada* completa. La *experiencia* extraída de cada jornada, que la hace distinta de la anterior, se rescata mediante la *memoria de retroalimentación*. Sólo guardamos en esta memoria la información de la última jornada, de forma que pasadas dos jornadas la información de la primera se pierde. La memoria de retroalimentación consiste de:<sup>12</sup>

- Memoria de recorrido continuo,  $MRC(a, t)$

$$\widehat{MRC}(a, t)_{n+1} = TRC(a, t)_n.$$

- Memoria de demora,  $M\chi(a, t)$

$$M\chi(a, t)_{n+1} = \chi(a, t)_n.$$

- Memoria de espacio libre,  $M\lambda_\emptyset(a, t)$

$$M\lambda_\emptyset(a, t)_{n+1} = \lambda_\emptyset(a, t)_n.$$

- Memoria de tiempo de recorrido,  $MTR(a, t)$

Tiene un uso específico al cual se debe su peculiar definición (ver3.6.9):

$$MTR(a, t + TR(a, t))_{n+1} = TR(a, t)_n.$$

Para evaluar las distintas opciones de rutas de un usuario de tipo  $\tau$  que llega a una intersección, asociamos a cada vértice y cada flecha un valor dinámico. En el caso de los vértices llamamos *situación* de  $v$  a la función que evalúa las opciones y la denotamos  $S_\tau(v, t)$ . La situación mide qué tan deseable es estar en el vértice  $v$  en el tiempo  $t$  para un usuario de tipo  $\tau$ . La definición de  $S_\tau(v, t)$  dependerá de qué se considere deseable y en qué medida. Proponemos la siguiente definición que se basa en la distancia relativa  $\widehat{d}_y^\tau(v, t)$ :

<sup>12</sup>Ya que la memoria de retroalimentación sólo guarda una jornada es claro que cada memoria se refiere a los datos de la jornada anterior. Por claridad incluimos en las definiciones los subíndices  $n$  y  $n + 1$ , mismos que indican la jornada a la que se refiere cada función. Por ejemplo, el espacio libre  $\lambda_\emptyset$  se almacena durante la jornada  $n$  para utilizarse en la siguiente jornada  $n + 1$  mediante la función  $M\lambda_\emptyset$ .

$$S_\tau(v, t) = \widehat{d}_y^\tau(v, t) \cdot \exp \left( \frac{1}{\|N^+(v)\|} \sum_{b \in N^+(v)} \frac{M\lambda_\emptyset(b, t) - \lambda_\emptyset(b, t)}{\lambda(b) \cdot \kappa(b)} \right) \quad (3.10)$$

donde  $\bar{\varsigma} = 0.9 \times \min_{\tau \in \mathfrak{I}} \{\varsigma_\tau\}$ .

Para la definición estándar de la efectividad, que se presenta a continuación, utilizamos además una situación futura, es decir, la predicción de la situación de un vértice dentro de cierto tiempo. La llamamos, *situación futura*, se denota  $S_\tau^*(w, b, t)$  y su definición es:

$$S_\tau^*(w, b, t) = \widehat{d}_y^\tau(w, t + TR(b, t)) \cdot \exp \left( \frac{1}{\|N^+(w)\|} \sum_{c \in N^+(w)} \frac{M\lambda_\emptyset(c, t) - \lambda_\emptyset(c, t)}{\lambda(c) \cdot \kappa(c)} \right) \quad (3.11)$$

donde  $b \in N^-(w)$  y  $\bar{\varsigma} = 0.9 \times \min_{\tau \in \mathfrak{I}} \{\varsigma_\tau\}$ .

En el caso de las flechas llamamos *efectividad* a la función de evaluación y la denotamos por  $E_\tau(b, t)$ . La efectividad mide las ventajas y desventajas que brinda una flecha, en el tiempo  $t$ , a un usuario de tipo  $\tau$ . Es un resumen de los criterios de selección de ruta, de modo que cada vehículo pueda elegir la mejor flecha en cada vértice formando la mejor ruta. En la figura 3.3 se muestran tres opciones típicas que buscamos poder comparar por medio de la efectividad: La ruta ideal por donde se puede circular libremente pero que no lleva al conductor a su destino, la ruta más directa al destino pero muy congestionada, y la intermedia entre ambas, que se puede transitar, lleva al destino pero no es directa.

La definición prototipo de la efectividad se basa en la situación de los vértices extremos de la flecha  $b = (v, w)$ , con vértices inicial y final  $v$  y  $w$ , respectivamente:

$$E_\tau(b, t) = \frac{S_\tau(v, t) - S_\tau^*(w, b, t)}{\widehat{TR}_\tau(b, t)} \quad (3.12)$$

donde  $\widehat{TR}_\tau$  es el tiempo de recorrido relativo (ver 3.6.8) y se considera, cuando se da el caso,  $\infty - \infty = 0$ .

Finalmente llega el momento en que el usuario decide qué flecha tomar. La efectividad evalúa las principales características de las opciones a escoger, por lo que es natural escoger aquella flecha de mayor efectividad. Sin embargo los usuarios no perciben ni valoran del mismo modo las características de un segmento. Para tomar en cuenta estas variaciones de percepción y valoración utilizamos el *parámetro aleatorio de comportamiento*  $\Psi$ , que determina la desviación del comportamiento base que consiste en elegir la flecha de mayor efectividad. De este modo, los usuarios percibirán la efectividad con algunas diferencias provocando diferentes comportamientos al momento de escoger una ruta.

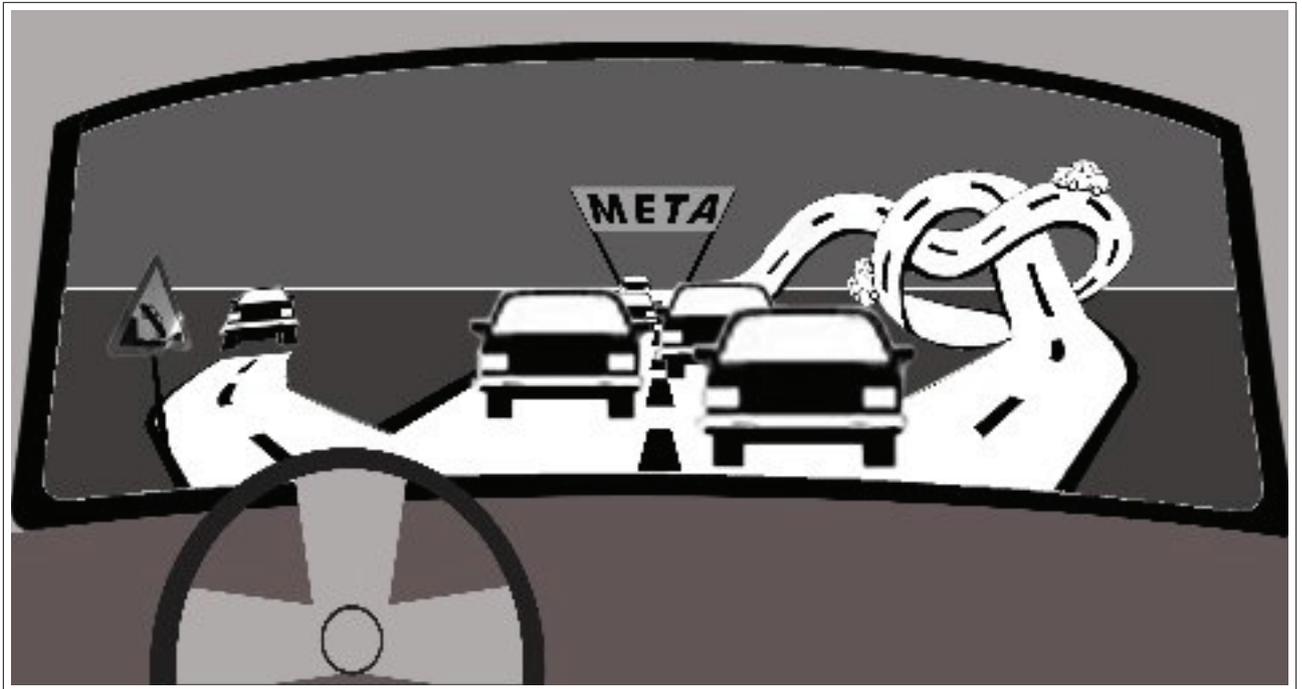


Figura 3.3: Cuál es la mejor vía o ruta?

Definimos  $E_\tau^\Psi(b, t)$  como la *efectividad percibida* de la flecha  $b$  por un usuario (o grupo de usuarios) particular de tipo  $\tau$ . La efectividad percibida se calcula como la desviación aleatoria de la efectividad:

$$E_\tau^\Psi(b, t) = E_\tau(b, t) \times \Psi.$$

Esta nueva percepción modifica ligeramente cada una de las efectividades para los distintos vehículos (aún del mismo tipo) de modo que, si dos flechas tienen efectividades similares, los conductores elegirán aleatoriamente entre ellas. Por otro lado, si una flecha tiene efectividad notoriamente mayor a las demás, la percepción particular de los conductores ampliará o disminuirá esta ventaja pero no logrará que desaparezca.

La *función de asignación* es la encargada de ejecutar la decisión de los vehículos, tomando en cuenta los distintos criterios y consideraciones descritas anteriormente, el prototipo de asignación es:

$$A_\tau(v, t) = c \tag{3.13}$$

donde  $c \in N^+(v)$  cumple que:

$$E_\tau^\Psi(c, t) = \max_{b \in N^+(v)} \{E_\tau^\Psi(b, t)\}. \tag{3.14}$$

Hasta ahora describimos el proceso por el cual los vehículos dentro de una flecha  $a$  la recorren, llevan a cabo el procedimiento de cola, y seleccionan, entre las flechas  $b \in N^+(v)$ , su siguiente flecha

$c$  en la que reiniciarán el ciclo. A continuación presentamos el proceso por el cual los vehículos entran y salen de las flechas, particularmente de la red cuando, por ejemplo, los conductores salen de su casa (origen) y llegan a su trabajo (destino).

### 3.3.3. Inicio y término de una jornada

Al principio de la jornada,  $t = 0$ , la red está vacía. De acuerdo a los horarios se lleva a cabo la *inyección de flujo*, es decir, inyectamos en el origen de cada capa  $\tau$ , según la información de la demanda, los vehículos que formarán el flujo dentro de las flechas.

La inyección de flujo se realiza de manera discreta en el tiempo y cantidad de vehículos dentro del lapso  $\Lambda$ , que se divide en pasos  $\pi$ . La *función de inyección*,  $I_\tau(a, t)$ , es la encargada de introducir el flujo a las flechas. El método de inyección introduce al monitoreo  $M_\tau$  los nuevos vehículos, o aquellos que provengan de otra flecha, como grupos de la forma  $(I_\tau(a, t), TRC(a, t))$ , donde  $I_\tau(a, t)$  es el tamaño del grupo y  $TRC$  el tiempo que tardarán en cruzar la flecha (ver 3.3.1). Si la flecha  $a$  en cuestión es una flecha auxiliar (i.e  $a \in \Xi$ ), la función de inyección da como resultado el número de vehículos de tipo  $\tau$  con origen  $x$  y destino  $y$  que inician su viaje en el paso  $t$ ; dato que es proporcionado por la demanda. Si la flecha  $a$  no es auxiliar, la función da el número de vehículos que, provenientes de otra flecha, arriban a la flecha  $a$ . Las flechas auxiliares  $h_\tau$  tienen longitud nula, por lo que su tiempo de recorrido es siempre cero, de manera que los grupos inyectados al monitoreo son de la forma  $(I_\tau(h_\tau, t), 0)$  y pasan inmediatamente a la cola.<sup>13</sup>

Inyectando el flujo en las flechas auxiliares,  $h_\tau \in \Xi$ , mediante  $I_\tau(h_\tau, 0)$  comienza el primer ciclo de seguimiento de las flechas. Cada ciclo de seguimiento corresponde a un paso enumerado mediante la variable  $t$ . Las flechas auxiliares  $h_\tau \in \Xi$  no están exentas del protocolo de cola, de hecho, si un vértice se encuentra muy congestionado es posible que en  $h_\tau$  se forme una cola, lo que significaría que los vehículos no pueden salir de su origen.

Consideramos que un usuario tipo  $\tau$  llega a su destino cuando alcanza el extremo de una flecha cuyo vértice final sea el destino del usuario. Un grupo de vehículos con tales características en  $M_\tau$  desaparece del flujo en lugar de entrar al protocolo de cola en  $a$ . De esta forma la gráfica se irá vaciando conforme los vehículos lleguen a sus destinos. La jornada termina cuando se acaba el lapso  $\Lambda$  y, a menos que todos los vehículos se queden detenidos (paralizados) en un nudo de vértices, la gráfica se vacía en un tiempo finito posiblemente mayor a  $\Lambda$  (ver 3.6.1).

---

<sup>13</sup>No hacemos la inyección directamente a la cola por simplicidad, para no añadir otro procedimiento, puesto que la mayoría de las inyecciones son en flechas no auxiliares, donde sí es necesario inyectar en el monitoreo  $M_\tau$ .

### 3.4. Iteración de jornadas. Convergencia

El proceso por el cual construimos el patrón de flujos periódico se basa en la iteración, de forma que en cada ciclo las rutas de los diferentes usuarios vayan adaptándose al flujo y condiciones generales de la red. Lo que marca la diferencia entre dos jornadas consecutivas es la memoria de retroalimentación, recopilada en la primera y que sirve como parámetro de selección de rutas en la segunda. De esta forma la experiencia de una jornada ayuda a encontrar una mejor ruta en la siguiente, lo que llamamos el proceso de adaptación, el cual termina en la construcción de las costumbres de viaje de los vehículos.

El siguiente problema es determinar cuándo hemos alcanzado el patrón de flujos periódico de la jornada que simulamos. Para esto llevamos a cabo una evaluación al término de cada jornada con datos recopilados a lo largo de la misma. Estos datos constituyen la memoria de evaluación. Para nuestra función prototipo de evaluación utilizamos solamente un dato:

- Memoria de Flujo,  $MF(a, t, n) = F(a, t)_n$

donde el subíndice  $n$  señala el número de la jornada de la que se colecta el dato. Guardamos los datos de las últimas  $N$  jornadas, donde  $N$  es la *magnitud de muestra*.

La evaluación consiste en determinar qué tan estable es la jornada. La sucesión de jornadas puede converger a una jornada que no cambie mucho a pesar de iterarse repetidas veces (equilibrio acotado), o quizá a un ciclo de jornadas, distintas entre sí, que no cambie mucho (equilibrio cíclico). Una tercera opción es que tenga un comportamiento caótico, que sin importar cuantas veces se itere, las diferencias entre las jornadas sean siempre significativas (Ver 1.2).

Llamamos  $ES(n)$  a la *estabilidad*,  $n$  indica que se calcula al final de la  $n$ -ésima jornada. Proponemos el siguiente prototipo:

$$ES(n) = \frac{\sum_{j=n-N+1}^{n-1} \left( \sum_{a \in F(G^*), t \in \Lambda} \frac{|MF(a, t, j+1) - MF(a, t, j)|}{\max\{1, MF(a, t, j)\}} \right)}{\|F(G)\| \times \|\Lambda\| \times N} \quad (3.15)$$

Dicha evaluación se realiza para  $n \geq N$ , a modo de tener una muestra completa de jornadas. Utilizamos el *parámetro de estabilidad*  $\epsilon \in \mathbb{R}^+$  como referencia de estabilidad, de manera que una muestra de jornadas para las que  $ES(n) < \epsilon$  se considera estable. El valor de  $\epsilon$  lo determina el usuario del modelo de acuerdo al grado de precisión que requiera en la estabilidad.

Si la jornada es estable, es decir si  $ES(n) < \epsilon$ , la iteración de jornadas se detiene, resultando la jornada  $n$  como representativa del patrón de flujos del tránsito periódico vehicular urbano. Este evento marca el final de la simulación y el comienzo de la evaluación de los datos obtenidos.

### 3.5. Resultados

El resultado principal es el flujo periódico en cada paso (tiempo) de la jornada, tanto el global como los parciales para cada tipo de usuario. En el caso del flujo global se cuenta con los datos de las últimas  $N$  jornadas que componen la muestra de evaluación. Además, se obtiene para la última jornada la efectividad, situación y congestión de todas las flechas y vértices, respectivamente, para todos los pasos; el tiempo de recorrido continuo y demora para cada flecha y paso; y la evaluación de estabilidad.

Estos datos, sobre todo el flujo global y los parciales, caracterizan el tránsito de la última jornada. A partir de ellos es posible hacer diversas evaluaciones del tránsito, algunas de las cuales presentamos a continuación.

#### 3.5.1. Niveles de servicio

En el contexto del manual de capacidad de carreteras (HCM, ver 2.3.2), “el nivel de servicio arterial se define en función de la velocidad media de recorrido de todos los vehículos de paso sobre la arteria”. Concretamente el nivel de servicio se determina conforme a la velocidad a flujo libre<sup>14</sup> (que suponemos igual a la velocidad de proyecto), y la velocidad media de recorrido de acuerdo al siguiente cuadro<sup>15</sup>:

Cuadro 3.1: Niveles de Servicio Arterial.

	Clasificación Arterial		
	I	II	III
Intervalo de variación de la velocidad libre (km/h)	72.4 a 56.3	56.3 a 48.3	56.3 a 40.2
Velocidad libre normal (km/h)	64.4	53.1	43.4
Nivel de Servicio	Velocidad Media de Recorrido (km/h)		
A	$\geq 56.3$	$\geq 48.3$	$\geq 40.2$
B	$\geq 45.0$	$\geq 38.6$	$\geq 30.6$
C	$\geq 35.4$	$\geq 29.0$	$\geq 20.9$
D	$\geq 27.4$	$\geq 22.5$	$\geq 14.5$
E	$\geq 20.9$	$\geq 16.1$	$\geq 11.3$
F	$< 20.9$	$< 16.1$	$< 11.3$

<sup>14</sup>La velocidad a flujo libre distingue el tipo de vía en la clasificación del HCM, el cuadro 3.1 determina el nivel de servicio de la vía de acuerdo a su tipo y la velocidad media de recorrido que presente.

<sup>15</sup>Cuadro obtenido íntegramente de [MCC] pag 597.

La velocidad media de recorrido  $VMR$  se puede calcular mediante el tiempo de recorrido  $TR$ . Ya que el HCM considera un periodo de 15 minutos como el mínimo para el cual se presenta una circulación estable, calculamos primero un valor promedio  $TR_p$  de tiempo de recorrido en el número de pasos  $p$ , equivalentes a 15 min ( $p = \frac{15min}{\pi seg}$ ), y a partir de éste, el valor de la velocidad media de recorrido  $VMR(a, t_0)$  para un periodo de 15 min comenzando en el tiempo  $t_0$ :

$$TR_p(a, t_0) = \frac{1}{p} \sum_{t=t_0}^{t_0+p} TR(a, t);$$

$$VMR(a, t_0) = \min \left\{ \frac{1}{\varpi(a)}, \frac{1}{TR_p} \right\};$$

considerando  $\frac{1}{TR_p} = \infty$  si  $TR_p = 0$ .

El nivel de servicio de una arteria se determina de acuerdo a la hora de máxima demanda, que por definición coincide con la de menor velocidad. Definimos entonces el *nivel de servicio* de la flecha  $a$  a partir de la mínima velocidad media de recorrido  $VMR_{\min} = \min_{t_0 \in \Lambda} \{VMR(a, t_0)\}$  y la tabla 3.1.

### 3.5.2. Consumo de energía y emisión de contaminantes

El sistema de transporte es, en muchos de los casos, el que más energía consume. Por ejemplo, en la zona metropolitana del valle de México el sector transporte consume en promedio el 51 % (en el periodo 1990-2004) de la energía anual requerida por esta región, seguido del sector industrial con un 34 %, residencial con 14 % y de servicios con 3 %. En cuanto a emisión de contaminantes, los vehículos automotores son los principales productores (99.2 %) de monóxido de carbono (el contaminante más abundante en peso) y óxidos de nitrógeno (82.2 %), además de tener importantes contribuciones en la producción de partículas menores a  $2.5 \mu m$  (56.6 %) y bióxido de azufre (50 %) [IE].

El consumo de energía y emisión de contaminantes de un vehículo a lo largo de un recorrido dependen, por parte del vehículo: del tipo de tecnología y combustible que utilice, y del modelo de vehículo; por parte del recorrido: de la pendiente, congestión (espera, frenado y acelerado en colas), obstáculos y longitud; además de otros factores entre los que destacan las condiciones ambientales de temperatura, altitud y humedad.

Por medio de los resultados del *modelo*<sup>16</sup> es posible calcular las emisiones de contaminantes y el consumo de energía de ellos. Los factores ambientales se asumen homogéneos para toda la red vial o grandes zonas de ésta. Para tomar en cuenta los factores del vehículo: dividimos los tipos de vehículo considerando, además de las características básicas, el tipo de tecnología y combustible

<sup>16</sup>Enfatizamos *modelo*, que se refiere a la estructura teórica, para distinguirlo del modelo (año) de los vehículos.

utilizado; por otra parte consideramos los distintos modelos de cada vehículo tomando en cuenta la distribución porcentual del flujo de cada tipo de vehículo en los diferentes modelos.

Para realizar el cálculo utilizamos los *factores de emisión* y *factores de consumo*, que indican, por kilómetro de recorrido, tipo y modelo de vehículo, las emisiones de los contaminantes más frecuentes, así como el consumo de combustible. Denotamos  $F_E(\gamma, mod)$  a la función que resume los factores de emisión de un cierto contaminante, y  $F_C(\gamma)$  a la función que indica el consumo; ambas para un vehículo modelo  $mod$  y tipo  $\gamma$ . Suponemos que los tipos de vehículos necesarios para la emisión corresponden con los definidos en el *modelo*<sup>17</sup>, por lo que se requiere como información adicional la proporción por modelo para cada tipo de vehículo, que denotamos  $P(mod)$ .

Mediante estos datos, tanto el consumo como la emisión se calculan para cada flecha  $a$  durante una jornada (o sección de ésta) como:

$$Consumo(a) = \sum_{\substack{\tau \in \mathbb{I}, t \in \Lambda \\ mod \in MOD}} F_C(\gamma(\tau), mod) \times P(mod) \times \lambda(a) \times \bar{v}_\tau(a, t);$$

$$Emisión(a) = \sum_{\substack{\tau \in \mathbb{I}, t \in \Lambda \\ mod \in MOD}} F_E(\gamma(\tau), mod) \times P(mod) \times \lambda(a) \times \bar{v}_\tau(a, t)$$

donde  $MOD$  es el conjunto de modelos,  $\lambda(a)$  la longitud de la flecha  $a$  y  $\bar{v}_\tau(a, t)$  el número de usuarios de tipo  $\tau$  que salieron de la flecha  $a$  en el tiempo  $t$ . Para calcular la emisiones o consumo en un cierto recorrido o zona de la red, basta sumar los consumos de las flechas que componen el recorrido o la zona en cuestión.

Es posible además tomar en cuenta las emisiones extras por las pendientes en el recorrido así como el tiempo, aceleración y frenado ocasionadas por congestión, lo que requiere de factores de emisión más específicos (emisión en colas). La fórmula es esencialmente la misma excepto que los factores de emisión dependen de las características de cada flecha<sup>18</sup>, es decir consideramos funciones extendidas de emisión y consumo, a saber  $F_E(\gamma(\tau), mod, a)$  y  $F_C(\gamma(\tau), mod, a)$ , respectivamente.

$$Consumo(a) = \sum_{\substack{\tau \in \mathbb{I}, t \in \Lambda \\ mod \in MOD}} F_C(\gamma(\tau), mod, a) \times P(mod) \times \lambda(a) \times \bar{v}_\tau(a, t);$$

$$Emisión(a) = \sum_{\substack{\tau \in \mathbb{I}, t \in \Lambda \\ mod \in MOD}} F_E(\gamma(\tau), mod, a) \times P(mod) \times \lambda(a) \times \bar{v}_\tau(a, t).$$

<sup>17</sup>Los tipos considerados para emisión y consumo son generalmente camionetas de carga, vehículos de capacidad menor a 3 ton, de capacidad mayor a 3 ton, tractocamiones, autobuses y motocicletas

<sup>18</sup>Para calcular las emisiones de un vehículo y recorrido en particular simplemente se considera el modelo y tipo de vehículo en cuestión.

### 3.5.3. Índice de correlación de vías

Existen sistemas de monitoreo del tránsito en muchas de las metrópolis del mundo. Estos sistemas determinan el flujo en cruceros y vías estratégicos con la finalidad de extrapolar la información a toda la red vial, obteniendo un estado general del tránsito. Una de las técnicas de extrapolación consiste en establecer índices de correlación entre las vías<sup>19</sup>. Un índice alto de correlación entre dos vías significa que éstas tienen la misma cantidad de flujo, de manera que si una de ellas está monitoreada, entonces se puede asignar el mismo flujo a la otra con alta precisión.

Proponemos un índice  $\mathcal{I}(a, b, t)$  de correlación de  $a$  y  $b$  basado en la interacción del flujo con la estructura de la gráfica, es decir en la efectividad.

$$\mathcal{I}(a, b, t) = \frac{1}{\|\mathfrak{J}(a) \cup \mathfrak{J}(b)\|} \sum_{\tau \in \mathfrak{J}(a) \cup \mathfrak{J}(b)} \sqrt{\max\{0, E_\tau(b, t)\} \times \max\{0, E_\tau(a, t)\}}$$

donde  $\mathfrak{J}(c)$  es el conjunto de tipos de usuario  $\tau$  de la flecha  $c$ , es decir, aquellos que cumplen:  $c \in \Theta_\tau$ .

La efectividad está íntimamente relacionada con las rutas *efectivas*, las que unen el origen y destino del usuario en cuestión. Una flecha que no se encuentra en una ruta *efectiva* tendrá efectividad nula o negativa. La efectividad es mayor si la flecha está dentro de una ruta mínima que sea *efectiva*. Por lo tanto, la efectividad no sólo responde a las características individuales de la flecha sino al lugar que ocupa dentro de la gráfica. Dos flechas con índice de correlación alto deben tener, para cada tipo de vehículo  $\tau \in \mathfrak{J}(a) \cup \mathfrak{J}(b)$ , una efectividad alta, de modo que pertenecen a rutas de recorrido cercano al mínimo. Un vehículo de tipo  $\tau \in \mathfrak{J}(a) \cup \mathfrak{J}(b)$  preferirá por un margen pequeño una de las rutas sobre la otra<sup>20</sup>, margen que bien puede invertirse por el parámetro aleatorio de decisión o por el cambio en los flujos en pocos pasos. Ya que consideramos todos los tipos de vehículos que pasan por las dos flechas en cuestión, la distribución de flujos para ellas debe equilibrarse. De esta forma obtenemos un índice de correlación para cada tiempo  $t \in \Lambda$ .

Muchas veces no se cuenta con registros de las vías monitoreadas para todos los horarios sino solamente para algunos estratégicos (como algún momento en la hora pico o valle) sobre los cuales recae la estimación de los flujos para periodos amplios de tiempo. Para esos casos proponemos utilizar el siguiente índice que es una media, para un periodo dado  $\bar{\Lambda} \subseteq \Lambda$ , del índice de correlación anterior.

$$I_{\bar{\Lambda}(a,b)} = \frac{1}{\|\bar{\Lambda}\|} \sqrt{\prod_{t \in \bar{\Lambda}} \mathcal{I}(a, b, t)}.$$

<sup>19</sup>Un trabajo en este sentido es el [CA]

<sup>20</sup>Cuando un usuario  $\tau$  sólo puede utilizar una de las flechas, es decir  $\tau \in \mathfrak{J}(a)$  y  $\tau \notin \mathfrak{J}(b)$ , se tiene que  $E_\tau(b, t) = 0 \forall t$  de modo que la contribución de  $\tau$  al promedio será nula, disminuyendo el valor de  $\mathcal{I}(a, b, t)$ . Si hay muchos usuarios de este tipo, entonces el promedio se acercará a cero. Esto nos dice que, para que la correlación sea alta ( $\mathcal{I}(a, b, t) \sim 1$ ) los conjuntos  $\mathfrak{J}(a)$  y  $\mathfrak{J}(b)$  no deben ser muy distintos.

### 3.5.4. Tiempos de viaje y ruteo

Aplicando Dijkstra dinámico con la función  $TR$  obtenemos todas las rutas mínimas y el tiempo de recorrido mínimo asociado a ellas, desde cualquier vértice y hacia todos aquellos vértices alcanzables desde él por un camino dirigido.

Al aplicar Dijkstra dinámico con  $\widehat{TR}_\tau$  se obtienen las rutas óptimas en términos de su efectividad. Los criterios de efectividad se pueden modificar, terminado el algoritmo, cambiando las preferencias de usuario de manera que al correr Dijkstra dinámico se obtengan rutas óptimas para diferentes criterios. Con este método podemos encontrar, por ejemplo, las rutas más seguras, más agradables y en mejores condiciones.

## 3.6. Justificación

### 3.6.1. Término de jornada

Hay tres tipos de finales para una jornada: (1) Cuando la gráfica termina vacía porque todos los vehículos llegaron a sus destinos; (2) cuando la gráfica no se vacía al término del lapso pero lo hace si continuamos la jornada por un tiempo adicional finito; y (3) cuando el flujo está atorado y la gráfica no se vaciará en tiempo finito.

Para el tercer caso hay dos opciones básicas, en la primera el movimiento de la red se detiene completamente a partir de algún tiempo  $t$ ; en la segunda existen vehículos que deambulan por la red infinitamente. El caso en que el movimiento se detiene completamente corresponde a redes particularmente densas respecto al flujo y espacio total y/o excepcionalmente mal diseñadas para el tránsito, pues los vehículos deben de obstruirse unos a otros los caminos para llegar a sus destinos sin que ninguno pueda moverse ni un poco hacia ningún lado.

El segundo caso no tiene símil con la realidad, nuestra confianza en que no ocurra de forma estable en las jornadas es un acto de Fe, misma que depositamos en la precisión del modelo. Si el modelo se apega lo suficiente al fenómeno, las jornadas que de él se obtienen también lo harán. Es suficiente que se cumplan las desigualdades 3.17 para asegurar que los vehículos siempre se acercan a su destino y no deambulan infinitamente por la red (ver 3.6.12). El que se cumplan dichas desigualdades depende de la precisión de la situación  $S$  y ésta a su vez de la estabilidad de las jornadas, lo cual depende esencialmente de todas las variables, es decir, del modelo en general. He ahí, al final de la cadena, nuestra Fe en que el modelo es apropiado.

### 3.6.2. Preferencias de usuario

Un segmento básico con características estándar es aquel que los usuarios consideran normal, en buen estado físico (pavimentación, ancho de carriles 3.60), seguro aunque sin vigilancia especial, con alumbrado público, no panorámico pero limpio. Los parámetros  $c_1, c_2, \dots$  de cada flecha miden dichas características, son el conjunto  $Car = \{c_1, c_2, \dots\}$  de características de la misma. Para calibrarlos se establece la característica estándar como cero, la ideal como 1 y la ínfima como -1, de modo que  $c_i \in [-1, 1] \forall i$ . Estos parámetros cuantifican características cualitativas de las flechas por lo que quedan a criterio de quien recolecte los datos.

De forma análoga cada tipo de usuario  $\tau$  posee un conjunto  $Pref = \{p_1, p_2, \dots\}$  de preferencias. A la preferencia  $p_i \in Pref$  le corresponde la característica  $c_i \in Car$  de modo que los conjuntos están apareados y por lo tanto tienen la misma cardinalidad. Denotamos  $s$  a dicho cardinal, de modo que:

$$\|\{c_1, c_2, \dots, c_s\}\| = \|Car\| = s = \|Pref\| = \|\{p_1, p_2, \dots, p_s\}\|.$$

La relación entre la preferencia de usuario  $p_1$  y su correspondiente característica  $c_1$  de cada flecha es tal que, con respecto a un segmento estándar (es decir,  $c_1 = 0$ ) de tiempo de recorrido 1, el usuario prefiere tomar una ruta de característica  $c_1$  siempre que recorrerla no requiera más de  $(1 + c_1 \cdot p_1)\pi$ . La fórmula  $(1 + c_1 \cdot p_1)\pi$  es entonces la aproximación lineal sobre  $c_1$  entre la preferencia por un segmento estándar ( $c_1 = 0$ ) y un segmento ideal ( $c_1 = 1$ ).

Para obtener la preferencia por entorno urbano de un segmento básico, la pregunta que se hace al usuario sería de la forma: *Piense en una calle que le guste recorrer por su entorno y una que le parezca normal. Si por la vía normal hace usted una hora a su destino, ¿cuánto tiempo más estaría dispuesto(a) a tardarse para irse por la calle que le gusta?*

Un usuario que no conoce bien la red vial, aún utilizándola de forma habitual, tenderá a utilizar las vías principales aunque esten congestionadas o le hagan recorrer mucho más distancia. Utilizamos la preferencia por vías principales ( $pvp$ ) para considerar a estos usuarios. Incorporamos el tipo de vía como otra de las característica  $c_i$  en el conjunto  $Car$ . En este caso  $c_i$  sólo tomará valores discretos en el intervalo  $[0, 1]$  donde el 0 corresponde a las vías de menor infraestructura (secundarias), el 1 a las de acceso controlado y los otros tipos toman valores intermedios. La preferencia por vías principales  $pvp$  nos dice qué tan dispuesto está el usuario a tomar vías alternas. Un valor de  $pvp \simeq 0$  corresponde a un usuario que conoce y utiliza los atajos de la red vial; un valor  $pvp \simeq 1$  indica que el conductor prefiere utilizar, en la medida de lo posible, vías principales y de acceso controlado.

Para condiciones desfavorables  $c_1 < 0$  como la inseguridad, la pregunta sería del siguiente tipo: *Piense en una calle que le parezca insegura y una que le parezca normal. Si por la vía normal hace usted una hora a su destino cuánto tiempo tendría que ahorrarse para elegir la calle insegura?*

Si el tiempo que tiene que ahorrarse es mayor al tiempo que toma recorrer la vía normal es imposible que el usuario elija tomar la insegura. La respuesta se convierte a las unidades  $\pi$  para formar el parámetro buscado.

Por más bella que sea una vía, excelentes sus condiciones físicas, seguridad, etc. hay un máximo de tiempo que el usuario está dispuesto a ceder para satisfacer sus preferencias. La preferencia por un segmento ideal es la que marca este límite. Definimos  $P_I$  de forma que: el usuario prefiere tomar una ruta de características perfectas  $c_1 = 1, c_2 = 1, \dots, c_s$  siempre que recorrerla no requiera más de  $(1 + P_I)\pi$ .

La preferencia de una ruta intermedia, entre lo estándar e ideal, debe ser algo intermedio entre la suma de sus preferencias y la preferencia ideal, de forma que haya congruencia entre la preferencia por una sola característica ( $p_i$ ) y por todas ( $P_I$ ). La fórmula que proponemos para  $P_\tau$  cumple que:  $P_\tau = p_i \times c_i$  si todas las características son estándar, excepto la  $i$ -ésima; y  $P_\tau = P_I$  si todas las características son perfectas. De este modo  $P_\tau$  respeta las preferencias iniciales y la ideal.

La fórmula para  $P_\tau(a)$  fue construida de la manera más sencilla en la que se cumplieran dichas condiciones, es decir lo más lineal posible. El resultado es una fórmula lineal en dos partes para cada variable  $c_i$ , donde el índice indica las características de la flecha  $a$  en cuestión y varía en el conjunto  $i \in \{1, 2, \dots, s\}$ :

$$P_\tau(a) = \begin{cases} \left( \sum_{1 \leq i \leq s} c_i \times p_i \right) & \text{si } \sum_{1 \leq i \leq s} c_i < 1; \\ \left( \sum_{1 \leq i \leq s} c_i \left( p_i + \frac{P_I - Q}{n - 1} \right) \right) + \frac{Q - P_I}{n - 1} & \text{si } \sum_{1 \leq i \leq s} c_i \geq 1; \end{cases}$$

donde  $Q = \sum_{1 \leq j \leq s} p_j$ , la suma de todas las preferencias. Construimos la fórmula considerando solamente dos preferencias de usuario y sus correspondientes características de flecha ( $s = 2$ ). Con esta base extrapolamos una fórmula para tamaño arbitrario ( $s \in \mathbb{N}$ ). En la figura 3.4 se muestra la gráfica de la función  $P_\tau$  en el caso base.

A continuamos verificamos que la fórmula para  $P_\tau$  cumple con lo previsto para los casos que motivaron su definición, a saber: cuando sólo una característica es perfecta y las otras normales, y cuando todas las características son perfectas.

- Cuando  $c_1 = 1$  y  $c_i = 0 \forall i \neq 1$  (tomamos  $c_j$  con  $j = 1$  sin pérdida de generalidad), tenemos:

$$\begin{aligned} P_\tau(a) &= \left( \sum_{1 \leq i \leq s} c_i \left( p_i + \frac{P_I - Q}{n - 1} \right) \right) + \frac{Q - P_I}{n - 1} \\ &= p_1 + \frac{P_I - Q}{n - 1} + \frac{Q - P_I}{n - 1} = p_1. \end{aligned}$$

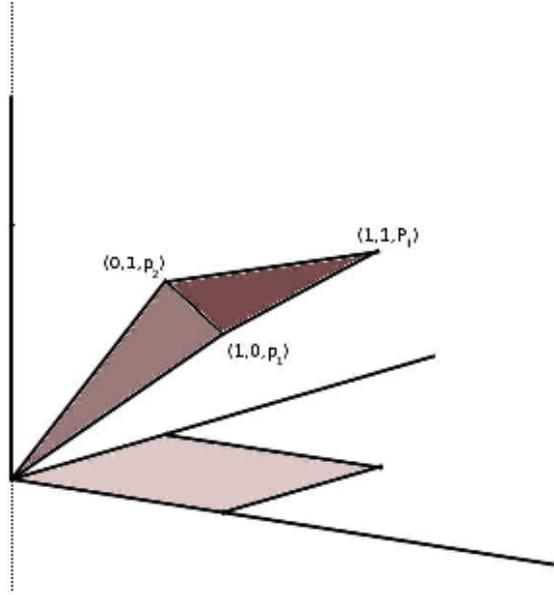


Figura 3.4: Preferencia Total  $P_\tau$ , caso base.

- Cuando  $\sum_{1 \leq i \leq s} c_i = 1$ :<sup>21</sup>

$$\begin{aligned}
 P_\tau(a) &= \left( \sum_{1 \leq i \leq s} c_i \left( p_i + \frac{P_I - Q}{n-1} \right) \right) + \frac{Q - P_I}{n-1} \\
 &= \left( \sum_{1 \leq i \leq s} c_i \times p_i \right) + \frac{P_I - Q}{n-1} + \frac{Q - P_I}{n-1} \\
 &= \left( \sum_{1 \leq i \leq s} c_i \times p_i \right).
 \end{aligned}$$

- Cuando  $c_i = 1 \forall i$  tal que  $1 \leq i \leq s$

$$\begin{aligned}
 P_\tau(a) &= \left( \sum_{1 \leq i \leq s} c_i \left( p_i + \frac{P_I - Q}{n-1} \right) \right) + \frac{Q - P_I}{n-1} \\
 &= \left( \sum_{1 \leq i \leq s} p_i \right) + n \times \frac{P_I - Q}{n-1} + \frac{Q - P_I}{n-1} \\
 &= Q + P_I - Q + \frac{P_I - Q}{n-1} + \frac{Q - P_I}{n-1} = P_I.
 \end{aligned}$$

<sup>21</sup>La función que calcula  $P_\tau$  es la aplicación de dos funciones continuas cuyos dominios ajenos cubren el dominio de  $P_\tau$ . Este caso muestra que en la frontera de los dominios, ambas funciones coinciden, por lo que  $P_\tau$  es también continua.

Cualitativamente, el valor obtenido  $P_\tau(a)$  es tal que el usuario prefiere tomar una ruta de características  $\{c_i(a)\}_{1 \leq i \leq s}$  siempre que recorrerla no requiera más de  $(1 + P_\tau(a))\pi$ .

### 3.6.3. Tiempo de recorrido continuo, $TRC(a, t)$

Para determinar  $TRC$  encontramos la velocidad de movimiento continuo, la cual se reduce de acuerdo a los obstáculos que los conductores encuentren en la vía, ya sean otros vehículos u obstáculos de diseño (ver 1.1). Los obstáculos de diseño están considerados por el modelo dentro de la velocidad a flujo libre, por lo que para esta función prototipo sólo tomaremos en cuenta como obstáculos a los otros vehículos de la vía.

Un conductor se ve obstaculizado por otro (ambos en movimiento) cuando su velocidad deseada es mayor que la velocidad de circulación del vehículo que tiene enfrente. El modelo (por ahora) no incluye distintas velocidades dado el tipo de vehículo, de modo que las velocidades deseadas de todos los vehículos son iguales entre sí y a la velocidad de proyecto. Por lo tanto, todo aquel que vaya por debajo de la velocidad de proyecto está siendo obstaculizado por un vehículo frente a él.

La distancia de frenado de un vehículo depende de la velocidad, a mayor velocidad es necesaria una mayor distancia para detener el vehículo. Cuando el flujo en una flecha es tal que la distancia entre los vehículos se reduce más allá de la necesaria para frenar a velocidad de proyecto, los conductores se ven obligados a bajar la velocidad hasta que corresponda con la distancia entre ellos y el vehículo que tienen enfrente.

De la tabla de distancias de frenado por velocidades en el código de carreteras del Reino Unido [HC] extrapolamos la siguiente fórmula que determina la distancia  $\mathcal{D}$  de frenado según la velocidad  $\mathcal{V}$ :

$$\mathcal{D}(\mathcal{V}) = \frac{5 \times \mathcal{V}^2}{(10 \times 1.609)^2} + \frac{\mathcal{V}}{1.609}$$

donde (1.609) aparece por la conversión de unidades, de millas a kilómetros. Encontramos su función inversa, que calcula la velocidad  $\mathcal{V}$  para la cual es necesaria una distancia  $\mathcal{D}$  de frenado:

$$\mathcal{V}(\mathcal{D}) = \left( -10 + \sqrt{100 + (20 \times 3.28)\mathcal{D}} \right) \times 1.609$$

donde (3.28) aparece por la conversión de unidades, de pies a metros.

La distancia  $\mathcal{D}$  que necesitamos es el espacio entre dos vehículos consecutivos, medido entre la defensa delantera del carro de atrás hasta la defensa trasera del carro de adelante. Para calcular  $\mathcal{D}$  encontramos el espacio promedio por vehículo dentro de la flecha  $a$ , ésto es, el *espacio libre*  $\lambda_\emptyset$  (el espacio total  $\lambda \times \kappa$ , menos el espacio ocupado en movimiento  $\lambda_M$  y cola  $\lambda_C$ ) entre el número de vehículos que están en movimiento ( $\mu$ , los que están en cola podemos suponer que no tienen espacio

entre ellos):

$$\mathcal{D} = \frac{\lambda_{\theta}(a, t)}{\mu(a, t)}$$

Entonces obtenemos un tiempo de recorrido de acuerdo a la distancia al vehículo de enfrente:

$$\text{tiempo de recorrido} = \frac{\lambda(a)}{\mathcal{V}\left(\frac{\lambda_{\theta}(a, t)}{\mu(a, t)}\right)}$$

A partir de cierta magnitud, el aumento en la distancia respecto al vehículo de enfrente no implica un aumento en la velocidad. Si un vehículo alcanza la velocidad deseada (de proyecto) no la aumenta aunque la vía en la que se encuentra esté vacía. De forma antagónica, cuando la distancia se reduce mucho la velocidad debe reducirse tanto que el vehículo entra en un proceso intermitente en el que se frena por completo hasta que la distancia aumenta y posteriormente avanza reduciéndola hasta que tiene que volver a detenerse. Dicho proceso no corresponde ya a un avance sostenido sino a una cola, en la que el movimiento depende fuertemente del desahogo de la cola más que de las distancias entre los vehículos. Por estas razones introducimos, en la función prototipo, dos parámetros que acotan la velocidad de recorrido continuo: la velocidad a flujo libre  $v(a)$  y el tiempo máximo de recorrido continuo  $\varpi(a)$ , de modo que definimos:

$$TRC(a, t) = \min \left\{ \varpi, \max \left\{ \frac{\lambda(a)}{v(a)}, \frac{\lambda(a)}{\left(-10 + \sqrt{100 + (20 \times 3.28) \frac{\lambda_{\theta}(a, t)}{\mu(a, t)}}\right) \times 1.609} \right\} \right\}.$$

La velocidad a flujo libre induce un tiempo de recorrido mínimo que acota a  $TRC$  por abajo, mientras que el tiempo máximo de recorrido continuo es una cota superior que separa el movimiento continuo del discontinuo propio de las colas.

#### 3.6.4. Función de demora, $\chi(a, t)$

Es el número de pasos estimados que durará el procedimiento de cola de un vehículo recién integrado a ella. Su definición prototipo es

$$\chi(a, t) = \begin{cases} \frac{\Upsilon(a, t-1)}{\iota(a, t-1)} & \text{si } \iota(a, t-1) > 0; \\ \infty & \text{si } \iota(a, t-1) = 0; \end{cases}$$

donde:

- $\Upsilon(a, t)$  es el número de vehículos que se encuentran haciendo cola en la flecha  $a$  al final del paso  $t$ .

- $\iota$  es la intensidad de paso.
- $\overline{\infty}$  es un valor que nos sirve para distinguir el origen del valor infinito en  $\widehat{TR}$ : en el caso de  $\overline{\infty}$  se debe a la función  $\chi$  mientras que  $\infty$  proviene de las preferencias de usuario (ver 3.3.2). Cumple lo siguiente:
  - Si consideramos los reales extendidos  $\mathbb{R}^* = \mathbb{R} \cup \infty$ , entonces  $\mathbb{R}^* \simeq \mathbb{R} \cup \overline{\infty}$ . De este modo las operaciones de  $\overline{\infty}$  con cualquier otro  $r \in \mathbb{R}$  son aquellas de  $\mathbb{R}^*$  y no utilizamos operaciones entre  $\infty$  e  $\overline{\infty}$ .
  - $\overline{\infty} < \infty$ . Definimos así esta relación puesto que el valor de  $\chi$  puede volverse finito después de algunos pasos, lo que no pasa con el valor de  $1 + P_\tau$  que es constante en el tiempo<sup>22</sup>.

La magnitud  $\frac{\Upsilon(a,t-1)}{\iota(a,t-1)}$  corresponde al tiempo que tardaría un vehículo en pasar por la cola si en cada paso salieran  $\iota(a, t-1)$  vehículos de la flecha  $a$ , que es una estimación de la cantidad de vehículos que pueden cruzar el vértice en el tiempo  $t-1$ . Definimos  $\chi(a, t) = \overline{\infty}$  cuando  $\iota(a, t-1) = 0$  pues significa que no hay desahogo de la cola y, de seguir así, los vehículos en ella nunca saldrán.

### 3.6.5. Tiempo de recorrido, $TR(a, t)$

Es el tiempo total de recorrido de la flecha  $a$ , empezando en el tiempo  $t$ . Se calcula como el tiempo en movimiento más el tiempo de demora por cola. Dado que nuestra definición de demora puede dar saltos bruscos, es necesario asegurar que  $TR$  cumpla con la *condición de fila* requerida para el algoritmo dinámico de Dijkstra (ver A.2.1). Con este objetivo definimos  $\bar{t}$  como el último tiempo, anterior al actual  $t$ , en que el tiempo de recorrido fue finito, es decir  $\bar{t} \in \Lambda$  tal que  $\bar{t} < t$  y  $TR(a, \bar{t}) < \overline{\infty} < \infty$ . Utilizando lo anterior definimos  $TR = \chi + TRC$  si se cumple la condición de fila, y  $TR(a, t) = TR(a, \bar{t}) - (t - \bar{t})$ , el valor extremo de la condición de fila, en el otro caso<sup>23</sup>.

$$TR(a, t) = \begin{cases} TRC(a, t) + \chi(a, t) & \text{si } TR(a, \bar{t}) \leq TRC(a, t) + \chi(a, t) + (t - \bar{t}) \\ TR(a, \bar{t}) - (t - \bar{t}) & \text{si } TR(a, \bar{t}) > TRC(a, t) + \chi(a, t) + (t - \bar{t}) \end{cases}$$

<sup>22</sup>Esta consideración se utiliza en el algoritmo dinámico de Dijkstra a modo de saber si el valor de infinito de  $\widehat{TR}(a, t)$  implica que no debe considerarse la flecha en cuestión o si se trata de una obstrucción momentánea y la flecha se podrá utilizar en un tiempo posterior.

<sup>23</sup>De esta forma, omitiendo los valores de  $t$  en que  $TR(a, t) = \infty$ , la condición de fila se preserva para cualquier pareja de tiempos. Basta que se cumpla esta versión débil de la condición de fila para que funcione el algoritmo de Dijkstra. Ver A.2.2.

### 3.6.6. Congestión, $\rho(v, t)$

Entendemos la congestión como el resultado de un conflicto por atravesar el mismo espacio, lo que evaluamos para el vértice  $v$  comparando: el número de vehículos que pretenden atravesarlo y la operación que cada uno hará (de qué flecha de entrada a qué flecha de salida pretende cruzar), frente a las capacidades de las flechas de salida para recibir vehículos.

Definimos  $\lambda_\emptyset(a, t) = \lambda(a) \cdot \kappa(a) - \lambda_\zeta(a, t) - \lambda_M(a, t)$  como el espacio libre en la flecha  $a$  al tiempo  $t$  y,  $D_\lambda(a, t)$  como la *demanda de espacio* en  $a$  al tiempo  $t$ , es decir el espacio que ocupan los vehículos que en el tiempo  $t$ , haciendo cola en el vértice  $v$  (inicial de  $a$ ), se dirigen hacia  $a$ . Llamemos *conflicto en flecha* a la magnitud  $\frac{D_\lambda(a, t)}{\lambda_\emptyset(a, t)}$  que, por lo anterior, es el espacio requerido por unidad de espacio disponible, en la flecha  $a$  al tiempo  $t$ . Entre mayor es el espacio requerido respecto al espacio disponible de la flecha, mayor es el *conflicto en flecha*.

Consideramos una flecha de salida *despejada* si el espacio requerido es menor o igual que el disponible. En términos de la congestión de un vértice, una flecha muy despejada ( $D_\lambda(a, t) \gg \lambda_\emptyset(a, t)$ ) tiene el mismo impacto que una apenas despejada ( $D_\lambda(a, t) \simeq \lambda_\emptyset(a, t)$ ), de modo que definimos el *conflicto en flecha* = 1 para cualquier flecha despejada. Entonces, redefinimos el *conflicto en flecha* como

$$\text{conflicto en flecha}(a) = \min\left(1, \frac{D_\lambda(a, t)}{\lambda_\emptyset(a, t)}\right).$$

La congestión,  $\rho(v, t)$  es un factor crítico para determinar el número de vehículos, de cada una de las flechas de salida, que cruzarán el vértice  $v$ . Digamos que, sin tomar en cuenta la congestión del vértice, cruzarán  $X$  vehículos de  $a$ , entonces para tomar en cuenta la congestión establecemos como  $X \cdot \rho(v, t)$  el número de vehículos que cruzarán. Si la congestión es muy alta, cuando el conflicto en las flechas de salida es grande, buscamos (1)  $\rho(v, t) \simeq 0$  de modo que  $X \cdot \rho(v, t) \simeq 0$ , que significa que cruzarán pocos vehículos. Por otro lado, si la congestión es poca, cuando las flechas de salida están despejadas, requerimos (2)  $\rho(v, t) \simeq 1$  de modo que  $X \cdot \rho(v, t) \simeq X$  como si no hubiera cruceo que obstruyera la circulación.

Buscamos evaluar la congestión del vértice, por lo que tomamos en cuenta los conflictos de todas las flechas de salida. Por lo tanto, definimos

$$\rho(v, t) = \frac{1}{\|N^+(v)\|} \cdot \sum_{a \in N^+(v)} \frac{1}{\text{conflicto en flecha}(a)}$$

es decir, como el promedio de los valores inversos del *conflicto en flecha* de las flechas de salida. Formalmente:

$$\rho(v, t) = \frac{\sum_{a \in N^+(v)} \min\left(1, \frac{\lambda_\emptyset(a, t)}{D_\lambda(a, t)}\right)}{\|N^+(v)\|}.$$

Cuando  $D_\lambda(a, t) = 0$  para alguna  $a$ , no hay conflicto en la flecha  $a$ , por lo que tomamos  $\frac{\lambda_\theta(a, t)}{D_\lambda(a, t)} \simeq 1$ , de manera que  $\rho$  está siempre bien definido.

### 3.6.7. Intensidad de paso, $\iota(a, t)$

$$\iota(a, t) = \left[ 0.5 \cdot \rho(v, t) \cdot \pi \cdot \kappa(a) \sum_{b \in N^+(v)} \phi(a, b) \cdot \kappa(a, b) \right]$$

donde  $v$  es el vértice final de  $a$ .

El manual de capacidad de carreteras [MCC] establece para condiciones ideales (que resumiendo, consisten de un conjunto de características del flujo, los conductores y la vía, las cuales maximizan la capacidad de dicha vía), una *intensidad de paso ideal* de 1800  $v/h/c$  (vehículos por hora por carril) que equivale a 0.5  $v/s/c$  (vehículos por segundo por carril). Ya que un paso dura  $\pi$  segundos, la tasa equivale a  $0.5 \cdot \pi \cdot v/\pi/c$  (vehículos por paso  $\pi$  por carril). La *intensidad ideal* está calculada con preferencia de paso absoluta (luz verde continua, en caso de semáforo) y sin congestión.

Para cada operación, digamos ir de  $a$  hacia  $b$ , ponderamos la *intensidad de paso ideal* con el parámetro de congestión para los  $\kappa(a)$  carriles, obteniendo:

$$0.5 \cdot \pi \cdot \rho(v, t) \cdot \kappa(a)$$

el número de vehículos en la flecha  $a$  que podrían cruzar el vértice  $v$  durante el paso  $t$ . Sin embargo no es posible hacer todas las maniobras de una vía al mismo tiempo; consideramos entonces para cada maniobra (ir de  $a$  hacia alguna  $b \in N^+(v)$ ) la fracción del paso  $t$  que dicha operación tiene preferencia de paso y la fracción de la capacidad que está proyectada para ir hacia  $b$ , a saber  $\kappa(a, b)$  (ver 3.1.1). Obtenemos como resultado el número de vehículos de  $a$  que pueden pasar a  $b$  durante el paso  $t$ , que son:

$$0.5 \cdot \pi \cdot \rho(v, t) \cdot \phi(a, b) \cdot \kappa(a) \cdot \kappa(a, b).$$

Sumando el valor anterior para todas las posibles operaciones, o lo que es lo mismo para todas las flechas  $b$  de salida del vértice  $v$ , obtenemos el número de vehículos que pueden salir de la flecha  $a$  durante el paso  $t$ . En el prototipo tomamos la función piso puesto que el número de vehículos debe ser siempre un entero.

### 3.6.8. Distancia relativa $\widehat{d}_y^\tau$ y gráfica relativa $\widehat{G}_\tau$

Cada tipo de usuario percibe los tiempos de recorrido de las vías de forma distinta según sus gustos y preferencias. El modelo modifica la distancia de las vías de la red, mediante el tiempo de

recorrido relativo  $\widehat{TR}_\tau$ , para representar la geometría relativa a las preferencias del usuario. Si se piensa como longitud, el tiempo de recorrido relativo modifica la topología de la red adaptándola a la percepción del usuario, en términos de sus preferencias y del flujo en cada tiempo  $t$ .

Cuando las preferencias de usuario determinan que éste no tomará la flecha  $a$  por ningún motivo ( $P_\tau(a) \leq -1$ ), el tiempo de recorrido relativo se define como infinito ( $\widehat{TR}_\tau(a) = \infty$ ) de forma que se ignore la flecha en cuestión en lo sucesivo. Las flechas  $a$  de  $G$  que cumplen  $1 + P_\tau(a) > 0$ , constituyen el conjunto de flechas no ignoradas por el usuario tipo  $\tau$ . Definimos entonces el tiempo de recorrido relativo  $\widehat{TR}_\tau(a, t)$  de una flecha  $a$ :

$$\widehat{TR}_\tau(a, t) = \begin{cases} \frac{TR(a, t)}{1 + P_\tau(a)} & \text{si } 1 + P_\tau(a) > 0; \\ \infty & \text{si } 1 + P_\tau(a) \leq 0. \end{cases}$$

De esta forma las preferencias de los usuarios de tipo  $\tau$  se resumen en tiempos de recorrido, prefiriendo la flecha para la que dicha magnitud sea menor.

Al ignorar una flecha la topología de la red cambia. Definimos la *gráfica relativa*  $\widehat{G}_\tau$  como la gráfica inducida por los vértices de  $G_\tau$  para los que hay un camino dirigido hacia  $y$  por flechas no ignoradas. Los cambios en  $\widehat{TR}_\tau(a, t)$  al variar  $t$  no modifican el signo pero sí la magnitud, lo que representa un cambio geométrico, más no topológico, en la red.

La gráfica  $\widehat{G}_\tau$  representa la red vial como la percibe el usuario tipo  $\tau$ . Una ruta mínima en esta gráfica representa el camino preferido del usuario, dado el horario de salida. Los vértices que no aparecen en la red no son útiles para los objetivos de transporte del usuario que se resumen en trasladarse de  $x_\tau$  a  $y_\tau$  con las restricciones impuestas por sus preferencias.

La distancia relativa  $\widehat{d}_y^\tau(v, t)$  es el tiempo mínimo de recorrido en la gráfica relativa (utilizando  $\widehat{TR}_\tau$ ) desde el vértice  $v$  hacia el  $y$  empezando en el tiempo  $t$ .

### 3.6.9. Memoria de tiempo de recorrido $MTR$

Definimos la memoria de tiempo de recorrido de la flecha  $a$  en el tiempo  $t^* = t + TR(a, t)$  como:

$$MTR(a, t + TR(a, t)) = TR(a, t)$$

de forma que, si un vehículo sale de la flecha  $a$  en el tiempo  $t^*$ , entonces  $MTR(a, t^*)$  es el número de pasos (tiempo) que tardó dicho vehículo en su recorrido por  $a$ . Esta definición es necesaria para calcular la función *volteada*  $\tilde{tr}$  que se utiliza para obtener los valores de  $d_y^\tau$  y  $\widehat{d}_y^\tau$  mediante el algoritmo dinámico o escalonado de Dijkstra, como se explica en el apéndice A.

Es posible que haya valores  $q$  para los que no exista  $t$  tal que  $t + TR(a, t) = q$ . En estos casos el valor  $MTR(a, q)$  no está definido. Para solucionar este problema *completamos*  $MTR$  inductivamente de la siguiente forma:

$$MTR(a, q) = \begin{cases} \frac{\lambda}{v} & \text{si } q \leq \frac{\lambda}{v}; \\ MTR(a, q) & \text{si está definida para } q; \\ MTR(a, q - 1) + 1 & \text{si no está definida para } q. \end{cases} \quad (3.16)$$

### 3.6.10. Situación, $S_\tau(v, t)$

Ya que la gráfica  $\widehat{G}_\tau$  resume los deseos de transporte del usuario, para establecer la situación de un vértice buscamos características de este respecto a la gráfica, a su estructura. La fórmula que consideramos se basa en la distancia de éste al destino  $y_\tau$ . Esta elección prioriza, como atractivo de un vértice, el que pertenezca a una ruta conveniente del usuario. La situación de los vértices mejora (se acerca a 0) mientras se acerca al destino, siendo este último el más deseable.

La situación de la jornada  $n$  estaría entonces definida por la distancia relativa  $\widehat{d}_y^\tau(v, t)$  de dicha jornada. Sin embargo  $\widehat{d}_y^\tau(v, t)$  no se puede calcular hasta que la jornada termine, por lo que es necesaria una aproximación. La aproximación que sugerimos es precisamente la función prototipo de la situación y se define como:

$$S_\tau(v, t) = \widehat{d}_y^\tau(v, t) \cdot \exp \left( \frac{1}{\|N^+(v)\|} \sum_{b \in N^+(v)} \frac{M\lambda_\theta(b, t) - \lambda_\theta(b, t)}{\lambda(b) \cdot \kappa(b)} \right).$$

La distancia  $\widehat{d}_y^\tau(v, t)$  se calcula mediante el patrón de tiempos de recorrido de la jornada anterior  $MTR$ . Aproximamos el valor de la distancia en la jornada  $n$  mediante el cambio en los espacios libres de las flechas de salida del vértice  $v$ .

Cuando  $\lambda_\theta < M\lambda_\theta$ , las flechas de salida del vértice  $b$  están más saturadas que en la jornada anterior, por lo que el tiempo de recorrido de estas, y probablemente el de las subsecuentes en el camino al destino, aumentará respecto a la jornada anterior aumentando la distancia al destino. Cuando  $\lambda_\theta > M\lambda_\theta$  sucede lo opuesto, de forma que esperamos que el tiempo de recorrido al destino disminuya. Por lo anterior el factor de ajuste aumenta o disminuye el tiempo de recorrido estimado al destino de acuerdo a una estimación local<sup>24</sup> basada en la saturación de las flechas de salida del vértice  $v$ . El término de ajuste que proponemos es:

$$\exp \left( \frac{1}{\|N^+(v)\|} \sum_{b \in N^+(v)} \frac{M\lambda_\theta(b, t) - \lambda_\theta(b, t)}{\lambda(b) \cdot \kappa(b)} \right).$$

<sup>24</sup>y de repercusión local, pues sólo afecta la elección de la siguiente flecha a tomar.

Cuando  $\lambda_\emptyset < M\lambda_\emptyset$  se cumple que:

$$0 < \frac{M\lambda_\emptyset(b, t) - \lambda_\emptyset(b, t)}{\lambda(b) \cdot \kappa(b)} < 1$$

por lo que

$$1 < \exp\left(\frac{M\lambda_\emptyset(b, t) - \lambda_\emptyset(b, t)}{\lambda(b) \cdot \kappa(b)}\right) < e$$

por lo tanto este valor, por sí solo, aumenta el valor de la distancia. Análogamente, cuando  $\lambda_\emptyset > M\lambda_\emptyset$  tenemos que

$$1/e < \exp\left(\frac{M\lambda_\emptyset(b, t) - \lambda_\emptyset(b, t)}{\lambda(b) \cdot \kappa(b)}\right) < 0$$

por lo que el valor por sí solo disminuye la distancia. El prototipo final promedia el efecto de cada uno de estos valores.

### 3.6.11. Situación futura, $S_\tau^*(w, b, t)$

Es una aproximación de la situación del vértice dentro de  $TR(b, t)$  pasos, la situación en que lo encontrará un vehículo que en el momento  $t$  inicie el recorrido de la flecha  $b = (v, w)$ . Se construye con el mismo principio que la situación normal excepto por que los valores  $M\lambda_\emptyset$  y  $\lambda_\emptyset$  se calculan para el tiempo  $t$  pues no se tienen aún los del  $t + TR(b, t)$ . El resultado es:

$$S_\tau^*(w, b, t) = \widehat{d}_y^\tau(w, t + TR(b, t)) \cdot \exp\left(\frac{1}{\|N^+(w)\|} \sum_{c \in N^+(w)} \frac{M\lambda_\emptyset(c, t) - \lambda_\emptyset(c, t)}{\lambda(c) \cdot \kappa(c)}\right).$$

### 3.6.12. Efectividad, $E_\tau(b, t)$

Para determinar la efectividad de una flecha respecto al usuario tipo  $\tau$ , tomamos en cuenta el punto de vista del usuario mediante la gráfica  $\widehat{G}_\tau$ , buscando las características de la flecha en cuestión que permiten alcanzar el destino.

Consideramos para una flecha  $b = (v, w)$  el cambio en su posición del vértice  $u$  al  $v$ . Este cambio lo medimos de acuerdo al avance o mejora en la situación de dichos vértices:

$$avance = S_\tau(v, t) - S_\tau^*(w, a, t).$$

La diferencia de las situaciones corresponde al avance, en términos de tiempo de recorrido relativo, que resulta de cruzar la flecha. Éste avance es distinto al tiempo real que toma cruzar la flecha ( $TR(b, t)$ ). Si un vehículo toma una flecha que no lo dirige directamente hacia su destino el tiempo de recorrido relativo no corresponderá con el de avance, sino que será mayor. El tiempo relativo iguala el avance cuando la flecha es parte de un camino preferido.

Para normalizar el *avance* dividimos entre el tiempo de recorrido relativo de la flecha, que corresponde al *avance* por unidad de tiempo relativo:

$$avance/\pi = \frac{S_\tau(v, t) - S_\tau^*(w, b, t)}{\widehat{TR}_\tau(b, t)}.$$

Así medimos, de acuerdo a las preferencias, la ventaja de cada paso que se da recorriendo la flecha  $a$ . Por lo tanto, definimos la efectividad como:

$$E_\tau(b, t) = \frac{S_\tau(v, t) - S_\tau^*(w, b, t)}{\widehat{TR}_\tau(b, t)}.$$

La efectividad no se calcula para flechas no auxiliares que, por ser las únicas de longitud nula, son las únicas para las que  $\widehat{TR}_\tau$  es cero. Por lo tanto, podemos asegurar que la efectividad está siempre bien definida.

La efectividad, dentro del modelo, se encarga de que los conductores tomen *la mejor ruta* que vayan encontrando. Esto requiere evitar los rodeos innecesarios o retroceder en el camino hacia el destino. Para ésto es suficiente que, mediante la efectividad, los vehículos distingan si una flecha los acercará a su destino o no, de forma que puedan elegir siempre una flecha que los acerca evitando regresar al mismo lugar o alejarse de su destino. Siempre que la situación es igual a la distancia  $\widehat{d}_y^\tau$ , la definición de efectividad cumple que:

$$\begin{aligned} \widehat{d}_y^\tau(v, t) &> \widehat{d}_y^\tau(w, t + TR(b, t)) && \text{si } E_\tau(b, t) > 0; \\ \widehat{d}_y^\tau(v, t) &= \widehat{d}_y^\tau(w, t + TR(b, t)) && \text{si } E_\tau(b, t) = 0; \\ \widehat{d}_y^\tau(v, t) &< \widehat{d}_y^\tau(w, t + TR(b, t)) && \text{si } E_\tau(b, t) < 0; \end{aligned} \quad (3.17)$$

lo cual sucede si  $M\lambda_\emptyset(v) = \lambda_\emptyset(v)$ . Por lo tanto, los conductores tomarán la mejor ruta en la medida en que el espacio libre sea constante para distintas jornadas, lo cual está íntimamente relacionado con que el volumen de flujo sea estable. Estas condiciones no son necesarias para que los conductores no den rodeos, su importancia radica en la probabilidad de que se cumplan cuando se ha hecho un número suficiente de iteraciones de la jornada.

### 3.6.13. Asignación, $A_\tau(v, t)$

Selecciona, entre las flechas de salida  $N^+(v)$  del vértice  $v$ , la *mejor* flecha para llegar al destino  $y_\tau$ . La flecha seleccionada debe acercar a su destino a un vehículo que comience a recorrerla en

el tiempo  $t$  y, respecto a las preferencias de usuario y su apreciación de las características de las flechas, ser la mejor opción para el tipo de usuario  $\tau$  en cuestión.

Consideramos que las distintas percepciones de los usuarios, aún del mismo tipo, varían alrededor de una percepción media, de modo que obtenemos las evaluaciones de las distintas percepciones como variaciones de una evaluación estándar construida con la percepción media. Utilizamos la efectividad como la evaluación estándar.

El parámetro aleatorio de comportamiento  $\Psi$  es un valor aleatorio que marca la variación del usuario respecto a la percepción media. El valor de  $\Psi$  no depende del tiempo, es un valor aleatorio que cambia cada vez que se utiliza y, si se observan los valores de varias aplicaciones, tiene una distribución normal con 1 como media y varianza  $\sigma^2 > 0$  fija (su valor es pequeño en condiciones apropiadas), cuando  $\Psi \sim N(1, \sigma^2)$ .

Seleccionamos la *mejor* flecha utilizando  $E_\tau^\Psi(b, t)$  como la evaluación particular de la flecha  $b$  por un usuario (o grupo de usuarios):

$$E_\tau^\Psi(b, t) = E_\tau(b, t) \times \Psi.$$

De esta manera, cada vehículo (o grupo de vehículos) percibirá la efectividad de la flecha  $b$  como  $E_\tau^\Psi(b, t)$  y elegirá la que entonces le parezca más eficiente. Por lo tanto, como función prototipo para la asignación de la siguiente flecha destino tenemos:

$$A_\tau(v, t) = c$$

donde  $c \in N^+(v)$  cumple que:

$$E_\tau^\Psi(c, t) = \max_{b \in N^+(v)} \{E_\tau^\Psi(b, t)\}.$$

La definición de  $A_\tau$  no asegura que los vehículos se acerquen a su destino cada vez que toman una nueva flecha, por lo que es posible que las rutas que los vehículos describen contengan rodeos innecesario o incluso que pasen más de una vez por el mismo vértice. Dada su definición prototipo la función de efectividad de las flechas disminuye cuando el flujo en ellas aumenta. Como el prototipo de  $A_\tau$  asigna la flecha con mayor efectividad, uno de los resultados es la asignación del flujo a las flechas menos saturadas. Esta dinámica tiende a *llenar* la red (o la parte de ésta que es útil para el tránsito) de manera homogénea de acuerdo a las características de sus vías. Una vez *llena*, los cambios en el patrón de flujos de la red corresponden a la composición del flujo en cada vía, más que al flujo neto.

Como se menciona en 3.6.12, la efectividad no corresponde con el acercamiento al destino cuando las cantidades de flujo en las flechas varían mucho de una jornada a otra, de manera que, cuando la red se *llena*, la efectividad corresponde con más precisión al *avance* hacia el destino. Los rodeos innecesarios se deben a la falta de correspondencia de la asignación con el acercamiento al destino, por lo que, cuando la red se *llena*, estos rodeos deben desaparecer.

### 3.6.14. Estabilidad, $ES(n)$

Para determinar la estabilidad consideramos las diferencias de flujo entre las  $N$  jornadas que forman la muestra. Escogimos el flujo por ser la magnitud que refleja de manera más directa la situación del tránsito. En el prototipo consideramos las diferencias entre jornadas consecutivas y las sumamos para considerar la variación total, desde la primera hasta la última jornada de la muestra. La diferencia entre dos jornadas consecutivas la definimos como la suma de las diferencias (en valor absoluto) de los flujos en cada flecha y cada tiempo  $t$ :

$$aumentado = \frac{\|MF(a, t, n+1) - MF(a, t, n)\|}{\max\{1, MF(a, t, n)\}}.$$

Dividimos entre  $\max\{1, MF(a, t, n)\}$  para obtener una diferencia proporcional entre los flujos de las dos jornadas (el 1 es para evitar indeterminación). Elegimos el flujo de la primera de las dos para que nos diga en qué proporción aumentará el flujo. Tomando en cuenta todas las aristas y tiempos:

$$aumentado\ agregado = \sum_{a \in F(G^*), t \in \Lambda} \frac{\|MF(a, t, n+1) - MF(a, t, n)\|}{\max\{1, MF(a, t, n)\}}. \quad (3.18)$$

Sumando la diferencia de todas las jornadas consecutivas de la muestra obtenemos:

$$aumentado\ total = \sum_{j=N-n+1}^{n-1} \left( \sum_{a \in F(G^*), t \in \Lambda} \frac{\|MF(a, t, j+1) - MF(a, t, j)\|}{\max\{1, MF(a, t, j)\}} \right).$$

Finalmente, promediamos el aumento total para obtener un *aumento agregado promedio* por flecha y tiempo, para toda la muestra, el cual constituye la estabilidad:

$$ES(n) = \frac{\sum_{j=n-N+1}^{n-1} \left( \sum_{a \in F(G^*), t \in \Lambda} \frac{|MF(a, t, j+1) - MF(a, t, j)|}{\max\{1, MF(a, t, j)\}} \right)}{\|F(G)\| \times \|\Lambda\| \times N}.$$

## Capítulo 4

# Implementación del modelo / Diseño del simulador

Este capítulo resume el modelo del capítulo anterior en un algoritmo que simula la primera relación de las variables principales (ver 1.1). Las entradas del algoritmo consisten de un objeto gráfica, que denotamos  $G$ , cuyos vértices y flechas, también objetos, contienen las características de los segmentos y cruceros que representan, si vemos la gráfica  $G$  como la red vial. Además el objeto gráfica  $G$  contiene la lista de usuarios, cada uno de los cuales tiene como datos de entrada sus preferencias, origen, destino y tipo de vehículo. La salida principal representa el patrón de flujos periódico y consiste de los flujos en cada flecha para cada paso de la jornada. Se obtienen también otros valores relacionados con el desarrollo de la jornada como la efectividad  $E_\tau(a, t)$  de las vías, la congestión  $\rho(v, t)$  de intersecciones y la distancia relativa a cada usuario  $\hat{d}_y^\tau(v)$ . A partir de los resultados y mediante procedimientos adicionales, obtenemos valores que comúnmente se utilizan para evaluar la situación del tránsito, como el nivel de servicio de una flecha (vía), el consumo de energía y la producción de contaminantes de la jornada.

Presentamos el algoritmo en forma de pseudo-código, el cual está pensado para programarse en un lenguaje orientado a objetos como lo es C++. El conjunto de funciones del modelo tiene una estrecha relación con los atributos de los objetos en este capítulo, sin embargo hay funciones del modelo que no aparecen explícitamente o que, dentro del pseudo-código son variables locales. Del mismo modo hay atributos cuya utilidad es simplemente operacional y no tienen relación directa con ninguna función del modelo; o atributos que, en diferentes partes del pseudo-código, toman el papel de distintas funciones del modelo.

Los parámetros del simulador van desde la unidad de tiempo  $\pi$  y el parámetro de evaluación  $\epsilon$ , hasta funciones completas, las llamadas funciones especiales como  $TRC$ ,  $\chi$ ,  $E_\tau$ , lo que complica la fase de ajuste de parámetros del modelo pero amplía las posibilidades de uso del mismo. La

implementación del simulador requiere, además de la programación del pseudo-código y el ajuste de parámetros, de una interfase de usuario que permita a éste manipular libremente las entradas y funciones del simulador, a modo de utilizar distintas estrategias que le permitan obtener los resultados que requiera. Entre estas posibilidades están: modificar la red vial y sus características, a modo de evaluar el efecto de ampliaciones, políticas viales (quitar vueltas a la izquierda en Insurgentes, por ejemplo); modificar la demanda, a modo de evaluar cambios en usos de suelo, movilidad poblacional o comercial; cambiar una ruta, compuesta de una sucesión de calles (flechas), por una sola flecha, simplificando el modelo a rutas (o segmentos de rutas); incluir variables de monitoreo, a modo de saber cuántos vehículos cruzan cada cruce por hora o la velocidad media en una ruta, por ejemplo.

#### 4.1. Convenciones del pseudo-código:

Las convenciones del pseudo-código que presentamos están basadas en el libro “Introduction to Algorithms” [CT] aunque no al pie de la letra pues hicimos varias modificaciones de acuerdo a las necesidades de este trabajo. Son las siguientes:

- El símbolo “▷” indica que el resto de la línea es un comentario.
- La tabulación (sangría) indica estructura de bloque. Tomamos como ejemplo las siguientes Instrucciones para subir una escalera<sup>1</sup> [CJ]:
  1. Agachándose y
  2. poniendo la mano izquierda en una de las partes verticales, y
  3. la derecha en la horizontal correspondiente ▷ se está en posesión
  4. ▷ momentánea de un peldaño o escalón.
  5. Las escaleras se suben de frente
  6. La actitud natural consiste en:
    7. mantenerse de pie,
    8. los brazos colgando sin esfuerzo,
    9. la cabeza erguida, y
    10. respirando lenta y regularmente.
  11. Basta repetir los siguientes movimientos hasta encontrarse con el final de la escalera.
  12. Comienza por levantar lo que llamaremos pie
  13. se recoge la parte equivalente de la izquierda y

---

<sup>1</sup>Fragmentos del texto “Instrucciones para subir una escalera” de Julio Cortázar

14. llevándola a la altura del pie:
  15. se le hace seguir hasta colocarla en el segundo peldaño.
  16. Se sale de la escalera fácilmente:
  17. con un ligero golpe de talón que la fija en su sitio.
- Los constructores de ciclos **while** y **for**, y los constructores condicionales **if** y **else** tienen la misma interpretación que en Pascal.
  - Una asignación múltiple de la forma  $i \leftarrow j \leftarrow k$  asigna a ambas variables  $i, j$  el valor de la expresión  $k$ ; debe ser tratada como equivalente a la asignación  $j \leftarrow k$  seguida de  $i \leftarrow j$ .
  - Las variables locales se declaran al inicio de cada método o procedimiento indicando el nombre de la variable y su tipo. Por ejemplo:

variable  $n \in \mathbb{N}$

indica que la variable  $n$  es un número natural.

- Los elementos de una lista o arreglo se acceden especificando el nombre del arreglo seguido por los índices ordenados, cada uno entre corchetes, que determinan el lugar del elemento dentro del arreglo. Por ejemplo,  $NacimientosPorMes[] []$  es un arreglo de números naturales donde el primer índice indica el mes y el segundo el año; es decir, si en junio de 1986 hubo 211'924 nacimientos, tenemos  $NacimientosPorMes[6][1986] = 211'924$ .

Una lista con dos o más índices es una lista múltiple. Retomando el ejemplo,  $NacimientosPorMes[*][1987]$  es una lista con índice  $*$  de los nacimientos en 1987 y, como tal, podemos llevar acabo la asignación

$$NacimientosPorMes1987[*] \leftarrow NacimientosPorMes[*][1987]$$

de forma que  $NacimientosPorMes1987[2] = 209'794$ , es el número de nacimientos en febrero de 1987 [INEGI].

- Para declarar una lista o arreglo basta indicar el nombre de éste seguido por las parejas de corchetes “[ ]”, que indican las variables que indexan el arreglo o lista, y el tipo de datos que contiene. En el ejemplo anterior la declaración sería:

$$\text{variable } NacimientosPorMes[] [] \in \mathbb{N}$$

Una lista es una arreglo de un solo parámetro. Por ejemplo, la lista de profesores de matemáticas:

variable  $ListaProfesores[] \in \mathbf{PROFESOR}$

$ListaProfesores[1] = \text{Javier Páez}$   
 $ListaProfesores[2] = \text{Hortensia Galeana}$   
 $ListaProfesores[3] = \text{Emilio Lluis Riera}$   
 $ListaProfesores[4] = \text{Luis Montejano}$   
 $ListaProfesores[5] = \text{Alejandro Bravo}$   
 $ListaProfesores[6] = \text{Carlos Torres}$   
 $ListaProfesores[7] = \text{Rafael Rojas}$   
 $ListaProfesores[0] = \text{El Roli}$

- Los datos compuestos se organizan en **objetos** que se componen de atributos o campos y métodos. Un atributo particular se accede usando el nombre del atributo seguido por el nombre de su objeto entre paréntesis. Un método es un procedimiento intrínseco al objeto, aunque puede recibir datos externos como parámetros. Para llamar un método se utiliza el nombre de éste junto con el nombre del objeto entre paréntesis y la lista de parámetros, también entre paréntesis. Siguiendo el ejemplo anterior: Javier Páez es un objeto de tipo **PROFESOR** con atributos: **cubículo**, **actividad** y **grado**, de forma que  $\text{cubículo}(\text{Javier Páez}) = 005$ ,  $\text{actividad}(\text{Javier Páez}) = \text{Docencia}$  y  $\text{grado}(\text{Javier Páez}) = \text{Doctor}$ . Un método de **PROFESOR** puede ser  $\text{SABÁTICO}()$ , que se encarga de actualizar los datos del profesor en relación al año sabático (donde el número de ocurrencias de sabáticos es parámetro del método). Al ejecutar  $\text{SABÁTICO}(\text{Javier Páez})(2)$ , obtenemos:  $\text{cubículo}(\text{Javier Páez}) = \emptyset$ ,  $\text{actividad}(\text{Javier Páez}) = 2^{\text{do}}$  año sabático y  $\text{grado}(\text{Javier Páez}) = \text{Doctor}$ .
- Una variable representando un arreglo u objeto es tratada como un puntero a los datos que representan el arreglo u objeto. Para todos los atributos  $f$  de un objeto  $x$ , asignar  $y \leftarrow x$  provoca  $f(y) = f(x)$ . Más aún, si asignamos  $f(x) \leftarrow 3$ , obtenemos no sólo  $f(x) = 3$  sino también  $f(y) = 3$ . En otras palabras,  $x$  y  $y$  apuntan al (son el) mismo objeto después de la asignación  $y \leftarrow x$ .
- Los parámetros pasan a los procedimientos (subrutinas) *por valor*: La subrutina recibe su propia copia de los parámetros de forma que, si asigna un valor a un parámetro dentro de ella, el cambio no influye en la superrutina (el procedimiento que llama). Cuando un objeto pasa como parámetro a una subrutina, el puntero a los datos que representan el objeto es copiado, pero los atributos del objeto no. Por ejemplo, si  $x$  es un parámetro de una subrutina, la asignación  $x \leftarrow y$ , dentro de la subrutina, no es visible para la superrutina. La asignación  $f(x) \leftarrow 3$ , sin embargo, es visible.

## 4.2. Datos de entrada

El algoritmo requiere, como datos de entrada, el objeto **GRÁFICA** y los parámetros  $\sigma$ ,  $\epsilon$ ,  $N$ ,  $\pi$  y  $\Lambda$ . La construcción de **GRÁFICA** requiere de datos de la red vial y sus características, que se incorporan como atributos de los objetos **VÉRTICE** y **FLECHA**; datos de origen-destino de viajes por horario y tipo de usuario, que se incorporan como atributos del objeto **USUARIO**; y características de los vehículos como atributos del objeto **VEHÍCULO**. El objeto **GRÁFICA** contiene como atributo una lista de todos los **VÉRTICES** que la conforman, lo mismo que de **FLECHAS** y **USUARIOS**; los distintos **VEHÍCULOS** son atributos del tipo de **USUARIO** que los utiliza para transportarse.

Las memorias de retroalimentación son atributos de **VÉRTICE** y **FLECHA**, según corresponda, mientras que la memoria de evaluación es atributo de **GRÁFICA**. Los datos que no se requieren para el desarrollo o evaluación de las jornadas, sino para un análisis posterior de éstas, se almacenan en memoria externa a los objetos.

El algoritmo regresa el objeto **GRÁFICA** que resulta de la última iteración y, por separado, los datos extras que el implementador precise. Como ejemplo de la utilidad de los resultados presentamos tres aplicaciones que obtienen el nivel de servicio, la emisión de contaminantes y un parámetro de relación funcional entre vías y cruces de la red.

## 4.3. Objetos

Decimos  $\tau \in \mathbf{USUARIO}$  cuando el objeto  $\tau$  es un objeto de tipo **USUARIO**.

Dentro de un objeto los atributos son únicos por lo que, para utilizarlos, no es necesario poner el nombre del objeto entre paréntesis. El atributo *ind* aparece en todos los objetos excepto en los de tipo **GRÁFICA**,<sup>2</sup> sin embargo no hay ambigüedad debido a que, en definiciones fuera del objeto, se utiliza el nombre del objeto al que *ind* se refiere (por ejemplo  $ind(v)$ , donde  $v \in \mathbf{VÉRTICE}$ ); y en definiciones dentro del objeto (**VÉRTICE** por ejemplo) se utiliza sin indicar cuando se refiere a dicho objeto (**VÉRTICE**) e indicando cuando se refiere a otro objeto ( $ind(\tau)$  si se refiere a un objeto  $\tau \in \mathbf{USUARIO}$  mientras está dentro de **VÉRTICE**).

- **VÉRTICE**  $v$ , tiene como atributos:
  - $ind \in \mathbb{N}$ , índice del vértice que nos permite identificarlo.

<sup>2</sup>Comúnmente se dice que el atributo está sobrecargado, análogamente a un abuso de notación.

- $I[ind(\tau)][ind(a)] \in \mathbb{N}$ , la inyección de flujo de usuarios de tipo  $\tau$  en la flecha  $a$  de la exvecindad de  $v$ .
- $N^+[] \in \mathbb{N}$  lista de índices de flechas en la exvecindad  $N^+(v)$ .
- $d_x^\tau[ind(\tau)] \in \mathbb{N}$ , distancia desde los orígenes.
- $d_y^\tau[ind(\tau)] \in \mathbb{N}$ , distancia a los destinos.
- $d_y^\tau[ind(\tau)][t] \in \mathbb{N}$ , distancia dinámica desde el vértice  $v$ , hacia el destino del usuario  $\tau$ , iniciando el viaje en el tiempo  $t$ .
- $\hat{d}_y^\tau[ind(\tau)][t] \in \mathbb{N}$ , distancia relativa al usuario  $\tau$ , desde el vértice  $v$ , hacia el destino, iniciando el viaje en el tiempo  $t$ .
- $\phi[ind(a)][ind(b)] \in [0, 1]$ , la preferencia de paso de la flecha  $a$  hacia la  $b$  en el vértice  $v$  (donde  $a$  y  $b$  se encuentran en la invecindad y exvecindad de  $v$ , respectivamente).
- $e \in \{0, 1\}$ , etiqueta binaria<sup>3</sup>
- $D_\lambda[ind(a)] \in \mathbb{N}$ , demanda de espacio en la flecha  $a$ : el espacio lineal (suma de longitudes) que ocupan los vehículos de la invecindad de  $v$  que están esperando cruzar hacia la flecha  $a$ .
- $\rho \in \mathbb{R}^+$ , congestión del vértice  $v$ .
- $M\lambda_\emptyset[t] \in \mathbb{R}^+$ , memoria de espacio libre.

Métodos de **VÉRTICE**:

- CONSTRUYE- $N$
- CALCULA- $\rho$
- ASIGNA-DESTINO
- SELECCIONA-VEH

■ **FLECHA**  $a$

Atributos respecto a la gráfica:

- $ind \in \mathbb{N}$ , índice de la flecha que permite identificarla.
- $vini \in \mathbf{VÉRTICE}$ , el vértice inicial de  $a$ .
- $vfin \in \mathbf{VÉRTICE}$ , el vértice final de  $a$ .
- $\mathfrak{I}_a[] \in \mathbf{USUARIO}$ , la lista de usuarios de la flecha  $a$ .

Atributos respecto a las características del segmento básico que representa:

- $\lambda \in \mathbb{R}^+$ , longitud de la flecha.
- $\kappa \in \mathbb{N}$ , número de carriles.

---

<sup>3</sup>Es una variable independiente al modelo, necesaria para llevar a cabo algunos de los procedimientos en que participa el objeto **VÉRTICE**.

- $\kappa[ind(b)] \in \mathbb{N}$ , número de carriles destinados a tomar  $b$  desde  $a$ .
- $v \in \mathbb{R}^+$ , velocidad de proyecto.
- $\varpi \in \mathbb{R}^+$  tiempo máximo de recorrido continuo.
- $Car[] \in [-1, 1]$ , lista de condiciones de la vía, donde  $Car[i]$  es el valor de la característica  $c_i$  de la flecha (de modo que corresponda con los valores de la lista  $Pref[]$  del objeto **USUARIO**).
- $\vartheta \in \mathbb{N}$ , tipo de vía.

Atributos respecto al flujo:

- $F \in \mathbb{N}$ , flujo en la flecha  $a$ , en relación a la función  $F(a, t)$ .
- $E[ind(\tau)] \in \mathbb{R}$ , efectividad de la flecha  $a$  respecto al usuario  $\tau$ .
- $\zeta[s] \in \mathbb{N}$ ,  $s \in \{1, 2, 3\}$ , cola de usuarios en la flecha. La sublista  $\zeta[i]$  equivale a la  $i$ -ésima terna  $(m, ind(b), ind(\tau))$ , de modo que  $\zeta[1][i] = m$ ,  $\zeta[2][i] = ind(b)$  y  $\zeta[3][i] = ind(\tau)$ .
- $M[ind(\tau)][s] \in \mathbb{N}$ ,  $s \in \{1, 2\}$ , monitoreo de usuarios tipo  $\tau$  en la flecha. La sublista  $M[ind(\tau)][i]$  equivale a las parejas  $(m, \delta)$  (análogo a  $\zeta$ ).
- $\Upsilon \in \mathbb{N}$ , número de usuarios en la cola  $\zeta(a, t)$ .
- $\Upsilon[ind(\tau)] \in \mathbb{N}$ , número de usuarios de tipo  $\tau$  en la cola  $\zeta(a, t)$ .
- $\mu \in \mathbb{N}$ , número de vehículos en  $\bigcup_{\tau \in \mathcal{T}} M_\tau(a, t)$ .
- $\mu[ind(\tau)] \in \mathbb{N}$ , número de vehículos en  $M_\tau(a, t)$ .
- $\omega^- \in \mathbb{N}$ , intensidad de salida.
- $M\omega^- \in \mathbb{N}$ , memoria de intensidad de salida.
- $TRC \in \mathbb{N}$ , tiempo de recorrido continuo.
- $TR \in \mathbb{N}$ , tiempo de recorrido.
- $\widehat{TR}[ind(\tau)] \in \mathbb{R}$ , tiempo de recorrido relativo al usuario  $\tau$ .
- $MTR[t] \in \mathbb{N}$ , memoria de tiempo de recorrido.
- $P_\tau[ind(\tau)] \in \mathbb{R}$ , preferencia total del usuario tipo  $\tau$ .
- $\lambda_\zeta \in \mathbb{R}^+$ , suma de las longitudes de los vehículos en  $\zeta(a, t)$ .
- $\lambda_M \in \mathbb{R}^+$ , suma de las longitudes de los vehículos en  $\bigcup_{\tau \in \mathcal{T}} M_\tau(a, t)$ .
- $\lambda_\emptyset \in \mathbb{R}^+$ , espacio vacío en la flecha  $a$ .
- $M\lambda_\emptyset \in \mathbb{R}^+$ , memoria de espacio vacío en la flecha  $a$ .
- $\iota \in \mathbb{R}$ , intensidad de paso en la flecha  $a$ .

Métodos de **FLECHA**:

- **CALCULA-E**

- *CALCULA-TR*
- *MONITOREO-M*
- **VEHÍCULO**  $\gamma$  tiene como atributos:
  - $ind \in \mathbb{N}$ , índice.
  - $\varsigma \in \mathbb{R}$ , tamaño de vehículo
  - $\theta[] \in \mathbb{N}$ , tipos de vías aptas
  - $\theta^*[] \in \mathbb{N}$ , conjunto adicional de vías aptas.
- **USUARIO**  $\tau$ , tiene como atributos:
  - $ind \in \mathbb{N}$ , índice.
  - $vorig \in \mathbf{VÉRTICE}$ , vértice de origen.
  - $vdest \in \mathbf{VÉRTICE}$ , vértice destino.
  - $h_\tau \in \mathbf{FLECHA}$ , flecha auxiliar.
  - $veh \in \mathbf{VEHÍCULO}$ , tipo de vehículo.
  - $Pref[] \in [0, 1]$ , lista de valores de preferencias del usuario (donde  $Pref[j]$  es el valor de la preferencia  $p_j$  asociada a la característica  $c_j$ , de modo que corresponda con los valores de la lista **Car**[] del objeto **FLECHA**).
  - $PI$ , preferencia ideal.
  - $dem[t] \in \mathbb{N}$ , demanda, número de viajes del usuario  $\tau$  que inician en el tiempo  $t$ .

Métodos de **USUARIO**:

- *CALCULA- $P_\tau$*
- **GRÁFICA**  $G$ , tiene como atributos:
  - $V[] \in \mathbf{VÉRTICE}$ , lista de vértices.
  - $F[] \in \mathbf{FLECHA}$ , lista de flechas.
  - $\mathfrak{U}[] \in \mathbf{USUARIO}$ , lista de usuarios.
  - $\mathfrak{T}[] \in \mathbb{N}$ , lista de tipos de vía en la gráfica.
  - $\Gamma[] \in \mathbb{N}$ , lista de tipos de vehículo en la gráfica.
  - $MF[n][ind(a)][t] \in \mathbb{N}$ , memoria de flujo correspondiente a  $MF(n, a, t)$ , el flujo en la jornada  $n$ , de la flecha  $a$ , al tiempo  $t$ .
  - $\Delta[n] \in \mathbb{R}^+$ , *Aumento agregado* de la jornada  $n - 1$  a la  $n$  en términos de la fórmula (3.18).
  - $\Lambda \in \mathbb{N}$ , cota superior del lapso. Es decir el lapso, en el contexto del simulador, es el conjunto  $\{1, 2, \dots, \Lambda\}$ .

Métodos de **GRÁFICA**:

- CONSTRUYE- $\widetilde{TR}$

Dadas estas definiciones de los objetos, una **GRÁFICA** puede no corresponder a una gráfica en el sentido estricto, ya que por ejemplo, los vértices extremos de sus flechas podrían no estar en la lista de vértices de **GRÁFICA**. El objeto **GRÁFICA**, que el algoritmo recibe como principal dato de entrada, representa la red vial con sus características y además contiene a los usuarios que llevan a cabo los viajes, por lo que necesitamos que el conjunto de usuarios  $\mathfrak{I}_a$  de **FLECHA** corresponda con la lista general  $\mathfrak{I}$  de usuarios de **GRÁFICA**.

En este contexto, decimos que un objeto **GRÁFICA** es *correcto* si sus atributos cumplen las siguientes condiciones:

- $vini(F[j]) \in V, \forall j$
- $vin(F[j]) \in V, \forall j$
- $vorig(\mathfrak{I}[j]) \in V, \forall j$
- $vdest(\mathfrak{I}[j]) \in V, \forall j$
- $\theta^*(\mathfrak{I}[j]) \subset F$
- $\theta(\mathfrak{I}[j]) \subset \mathfrak{I}, \forall j$
- $\mathfrak{I}_a(F[j]) \subset \mathfrak{I}, \forall j$
- $\vartheta(a) \in \mathfrak{I}, \forall a \in F$

Los vértices y las flechas son independientes a las gráficas, éstas consisten de ellos, de manera que gráficas distintas pueden compartir el mismo vértice o la misma flecha, cada uno de los cuales tiene valores únicos, independientes de la gráfica, para sus atributos.

## 4.4. Algoritmo

Está formado por dos ciclos principales anidados. El ciclo denominado General que se encarga de la convergencia de las jornadas  $J_i$  y corresponde a la regla de iteración  $\mathfrak{S}$ ; y el denominado Jornada que calcula el patrón de flujos de cada jornada  $J_i$  y corresponde a la regla de iteración

$F$  (ver 3.2.2). Cada ciclo se forma de subrutinas (métodos y procedimientos), que conforman las secciones siguientes.

Dado un algoritmo, método o procedimiento  $\mathfrak{A}$  y  $n_1, n_2, \dots, n_k$  el número de datos de los  $k$  tipos de datos de entrada que  $\mathfrak{A}$  requiere, decimos que el tiempo de ejecución  $TE(\mathfrak{A})$  de  $\mathfrak{A}$  es de orden  $\mathcal{O}(f(n_1, n_2, \dots, n_k))$  si existen constantes  $c_1$  y  $c_2$  que cumplen:

$$c_1 \cdot f(n_1, n_2, \dots, n_k) \leq TE(\mathfrak{A}) \leq c_2 \cdot f(n_1, n_2, \dots, n_k)$$

donde  $f(n_1, n_2, \dots, n_k)$  es un polinomio en las variables  $n_1, n_2, \dots, n_k$ . Entonces, el tiempo de ejecución del simulador es de orden  $\mathcal{O}(\mathfrak{J} \cdot V^2 \cdot \Lambda^3)$ , donde  $V$  es el número de vértices de la gráfica  $G$ ,  $\Lambda$  el número de pasos en una jornada y  $\mathfrak{J}$  el número de usuarios distintos.

Las diferentes versiones y aplicaciones del algoritmo de Dijkstra para calcular las distancias son los factores determinantes al calcular el tiempo de ejecución del algoritmo general. El algoritmo puede funcionar utilizando como orientación la distancia dinámica o la relativa. Si utilizamos la dinámica, la orientación corresponde al tiempo de recorrido de la red de acuerdo al flujo. Si tomamos la distancia relativa, la orientación corresponde al tiempo de recorrido relativo a cada usuario, tomando en cuenta el flujo y las preferencias de los usuarios. El tiempo de ejecución utilizando la versión dinámica es de orden  $\mathcal{O}(V^2 \Lambda \mathfrak{J} + \Lambda^2 V)$ . En la versión que se presenta a continuación y para el cálculo del orden de ejecución, utilizamos la distancia relativa, sin embargo incluimos los procedimientos que se requieren para la versión dinámica.

Para calcular el orden del tiempo de ejecución tomamos en cuenta lo siguiente:

- Las exvecindades en la gráfica de una red vial son de cardinalidad acotada y, de hecho, pequeña, menor que 10; de forma que, con respecto al orden de un procedimiento, podemos considerarlas constantes.
- Por la misma razón, la proporción del número de flechas respecto al número de vértices puede considerarse constante. Al especificar el orden del tiempo de ejecución de cada método o procedimiento utilizamos el número de flechas  $\|F(G)\|$  sin embargo, en el cálculo final del orden, lo reemplazamos por el número de vértices  $V = \|V(G)\|$ .
- El número de vehículos dentro de una flecha es acotado, ya que la longitud y el número de carriles de ésta es constante. Por lo tanto, el número de vehículos en los monitoreos  $\{M_\tau\}_{\tau \in \mathfrak{J}_a}$  y en la cola  $\zeta$  puede considerarse constante.
- El número de preferencias de usuario y condiciones de la flecha es constante.

#### 4.4.1. CicloGeneral( $G, \pi, \Lambda, \epsilon, \sigma, N$ )

En el ciclo comprendido entre las líneas 8 y 59 construye las gráficas  $G_\gamma, G_\tau$  y  $G_\tau^*$  que se describen en la sección 3.1.2, la primera de forma explícita en la línea 11 y las últimas dos de forma implícita, utilizando los conjuntos de usuario de las flechas, entre las líneas 19 a 48 y 49 a 57, respectivamente.<sup>4</sup> El ciclo entre las líneas 84 y 86 corresponde a la iteración de jornadas de la sección 3.4.

1. variable  $i, j, g, n \in \mathbb{N}$
2. variable  $v \in \mathbf{VÉRTICE}$
3. variable  $a, h \in \mathbf{FLECHA}$
4. variable  $\gamma \in \mathbf{VEHÍCULO}$
5. variable  $\tau \in \mathbf{USUARIO}$
6. variable  $G_\gamma, \tilde{G} \in \mathbf{GRÁFICA}$
7.  $\triangleright$  Para cada vehículo  $\gamma$  de  $G$ :
8. **for**  $g \leftarrow 1$  **to**  $\mathbf{CARDINAL}(\Gamma[*])$
9.      $\gamma \leftarrow \Gamma[g]$
10.  $\triangleright$  Construye la gráfica  $G_\gamma$  con las vías aptas para  $\gamma$ .
11.      $G_\gamma \leftarrow \mathbf{Construye}G_\gamma(G, \gamma)$
12.  $\triangleright$  Para los vértices  $v$  de  $V(G_\gamma)$  contruye la invecindad y exvecindad de  $v$ .
13.     **for**  $j \leftarrow 1$  **to**  $\mathbf{CARDINAL}(V(G_\gamma)[*])$
14.          $v \leftarrow V(G_\gamma)[j]$
15.          $\mathbf{CONSTRUYE-N}(v)(G_\gamma)$
16.          $j \leftarrow j + 1$
17.  $\triangleright$  Para los usuarios  $\tau$  cuyo vehículo sea  $\gamma$ , construye la gráfica  $G_\tau$ :

---

<sup>4</sup>La gráfica  $G_\gamma$  se construye para poder correr sobre ella el algoritmo de Dijkstra y determinar la vecindad  $H_\tau$  (ver *capas de usuario* cap. 3.1.2). Las flechas y vértices de  $G_\gamma$  son independientes a la gráfica y los cambios hechos en ellos, tales como el calculo de su atributo  $\widehat{d}_y^x$  y la construcción del conjunto de usuarios  $\mathfrak{J}_a$ , también afectan a la gráfica  $G$ , dado que  $V(G_\gamma) \subset V(G)$  y  $F(G_\gamma) \subset F(G)$ . La gráfica  $G_\tau$  del capítulo 3 es aquella formada por las flechas para las que  $\tau \in \mathfrak{J}_a$  y los vértices extremos de éstas.

18.     **for**  $i \leftarrow 1$  to  $\text{CARDINAL}(\mathfrak{I}(G)[*])$
19.              $\tau \leftarrow \mathfrak{I}(G)[i]$
20.             **if**  $\gamma(\tau) = \gamma$
21.      $\triangleright$  Establece la distancia  $d_x^r[ind(\tau)]$  dentro de  $G_\gamma$  utilizando Dijkstra con
22.      $\triangleright$       $\frac{\lambda}{v}$  como función de costos sobre las flechas.
23.              $DJT(G_\gamma, \text{vorig}(\tau), \frac{\lambda}{v})$
24.      $\triangleright$  Establece la distancia  $d_y^r[ind(\tau)]$  dentro de  $G_\gamma$  volteandola, utilizando Dijkstra
25.      $\triangleright$      con  $\frac{\lambda}{v}$  como función de costos sobre las flechas y finalmente
26.      $\triangleright$      regresándola a su estado original.
27.              $\widetilde{G}_\gamma \leftarrow \text{VolteaG}(G_\gamma)$
28.              $DJT(\widetilde{G}_\gamma, \text{vdest}(\tau), \frac{\lambda(a)}{v(a)})$
29.              $G_\gamma \leftarrow \text{VolteaG}(\widetilde{G}_\gamma)$
30.             **for**  $j \leftarrow 1$  **to**  $\text{CARDINAL}(F(G_\gamma)[*])$
31.                  $a \leftarrow F(G_\gamma)[j]$
32.      $\triangleright$  Establece  $\tau$  como usuario de la flecha  $a$ , si los vértices extremos de la flecha están
33.      $\triangleright$      en una ruta que una el origen y destino del usuario  $\tau$  y no exceda en más de  $\sigma$  la
34.      $\triangleright$      longitud de la ruta más corta.
35.             **if**  $\begin{cases} d_x^r(\text{vini}(a))[ind(\tau)] + d_y^r(\text{vini}(a))[ind(\tau)] \leq d_x^r(\text{vdest}(\tau))[ind(\tau)] + \sigma \text{ y} \\ d_x^r(\text{vfin}(a))[ind(\tau)] + d_y^r(\text{vfin}(a))[ind(\tau)] \leq d_y^r(\text{vorig}(\tau))[ind(\tau)] + \sigma \end{cases}$
36.                  $\text{AÑADE}(\tau, \mathfrak{I}_a(a)[*])$
37.                  $j \leftarrow j + 1$
38.      $\triangleright$  Incorpora la flecha auxiliar del usuario  $\tau$  a la gráfica  $G$  inicializando los
39.      $\triangleright$      atributos que requiere y estableciendo  $\tau$  como usuario de ésta.
40.                  $h \leftarrow h_\tau(\tau)$
41.                  $\text{vfin}(h) \leftarrow \text{vorig}(\tau)$
42.                  $\lambda(h) \leftarrow 0$

- 
43.  $\kappa(h) \leftarrow 0$
  44. **for**  $j \leftarrow 1$  **to**  $\text{CARDINAL}(N^+(vfin(h))[*])$
  45.  $a \leftarrow N^+(vfin(h))[j]$
  46.  $\kappa(h) \leftarrow \kappa(h) + \kappa(a)$
  47.  $\phi(vfin(h))[ind(h)][ind(a)] \leftarrow 1$
  48.  $j \leftarrow j + 1$
  49.  $\varpi(h) \leftarrow 0$
  50.  $v(h) \leftarrow 1$
  51. **for**  $j \leftarrow 1$  **to**  $\text{CARDINAL}(Car[*])$
  52.  $Car[j] \leftarrow 0$
  53.  $j \leftarrow j + 1$
  54.  $\text{AÑADE}(\tau, \mathfrak{J}_a(h)[*])$
  55.  $ind(h) \leftarrow \text{CARDINAL}(F(G)[*]) + 1$
  56.  $\text{AÑADE}(h, F(G)[*])$
  57.  $\text{AÑADE}(\tau, \mathfrak{J}(h)[*])$
  58.  $i \leftarrow i + 1$
  59.  $g \leftarrow g + 1$
  60.  $\triangleright$  Calcula la preferencia total de las flechas, para cada usuario  $\tau$ .
  61. **for**  $i \leftarrow 1$  **to**  $\text{CARDINAL}(\mathfrak{J}(G)[*])$
  62.  $\tau \leftarrow \mathfrak{J}(G)[i]$
  63. **for**  $j \leftarrow 1$  **to**  $\text{CARDINAL}(F(G)[*])$
  64.  $a \leftarrow F(G)[j]$
  65.  $\text{CALCULA-}P_\tau(a)$
  66.  $j \leftarrow j + 1$
  67.  $i \leftarrow i + 1$
  68.  $\triangleright$  Para los vértices  $v$  de  $V(G)$  construye la invecindad y exvecindad de  $v$ .

```

69. for  $j \leftarrow 1$  to  $\text{CARDINAL}(V(G)[*])$ 
70.      $v \leftarrow V(G)[j]$ 
71.      $\text{CONSTRUYE-N}(v)(G)$ 
72.      $j \leftarrow j + 1$ 
73.  $n \leftarrow 0$ 
74.  $\triangleright$  Inicializa las memorias.
75. for  $t \leftarrow 1$  to  $\Lambda$ 
76.     for  $i \leftarrow 1$  to  $\text{CARDINAL}(F(G)[*])$ 
77.          $a \leftarrow F(G)[i]$ 
78.          $MTR(a) \left[ t + \left\lceil \frac{\lambda(a)}{v(a)} \right\rceil \right] = \left\lceil \frac{\lambda(a)}{v(a)} \right\rceil$ 
79.          $M\lambda_{\emptyset}(a)[t] \leftarrow \lambda(a) \cdot \kappa(a)$ 
80.          $i \leftarrow i + 1$ 
81.      $t \leftarrow t + 1$ 
82.  $\text{Construye-}\widehat{d}_y^+(G)$ 
83.  $\triangleright$  Itera las jornadas mientras el procedimiento de evaluación no determine convergencia.
84. while NOT  $\text{EvalConv}(G, n)$ 
85.      $n \leftarrow n + 1$ 
86.      $\text{JORNADA}(G, n)$ 
87.  $\triangleright$  Regresa la gráfica  $G$  con los datos de su última jornada.
88. return  $G$ 

```

#### 4.4.2. $\text{JORNADA}(G, n)$

Corresponde a la sección 3.3: las líneas 48 a 60 se basan en la subsección 3.3.1, las líneas 41, 45, 63 y 64 en la 3.3.2. Por último, los procedimientos relativos a la subsección 3.3.3 de inicio y término se llevan a cabo dentro de los ya mencionados.

1.  $\triangleright$  Variables locales:
2. variables  $i, t, j, k \in \mathbb{N}$
3. variables  $v \in \mathbf{VÉRTICE}$
4. variables  $a \in \mathbf{FLECHA}$
5. variables  $\tau \in \mathbf{USUARIO}$
6.  $\triangleright$  **Inicialización**
7.  $t \leftarrow 0$
8. **for**  $i \leftarrow 1$  to  $\text{CARDINAL}(F(G)[*])$
9.      $a \leftarrow F(G)[i]$
10.      $F(a) \leftarrow 0$
11.      $\zeta(a)[\ ] \leftarrow 0$
12.      $M(a)[\ ][\ ] \leftarrow 0$
13.      $\Upsilon(a)[\ ] \leftarrow 0$
14.      $\Upsilon(a) \leftarrow 0$
15.      $\mu(a)[\ ] \leftarrow 0$
16.      $\mu(a) \leftarrow 0$
17.      $\lambda_{\emptyset}(a) \leftarrow \lambda(a) \cdot \kappa(a)$
18.      $TRC(a) \leftarrow \frac{\lambda(a)}{v(a)}$
19.      $i \leftarrow i + 1$
20. **for**  $i \leftarrow 1$  to  $\text{CARDINAL}(V(G)[*])$
21.     **for**  $j \leftarrow 1$  to  $\text{CARDINAL}(N^+(v)[*])$
22.          $a \leftarrow N^+(v)[j]$
23.          $D_{\lambda}(v)[\text{ind}(a)] \leftarrow 0$
24.          $j \leftarrow j + 1$
25.      $v \leftarrow V(G)[i]$
26.      $\rho(v) \leftarrow 1$

27. *Calcula*  $-\iota(v)$
28.  $i \leftarrow i + 1$
29. **▷ Paso algorítmico:**
30. **▷** Para cada tiempo  $t$ :
31. **while**  $t \leq \Lambda$
32. **▷** Toma cada uno de los vértices  $v$ :
33. **for**  $i \leftarrow 1$  **to**  $\text{CARDINAL}(V(G)[*])$
34.  $v \leftarrow V(G)[i]$
35. **▷** Para las flechas  $b$  de su exvecindad:
36. **for**  $j \leftarrow 1$  **to**  $\text{CARDINAL}(N^+(v)[*])$
37.  $b \leftarrow N^+(G)[j]$
38. **for**  $k \leftarrow 1$  **to**  $\text{CARDINAL}(\mathfrak{I}_a(b)[*])$
39.  $\tau \leftarrow \mathfrak{I}_a(b)[k]$
40. **▷** Establece el parámetro  $E_\tau(a)[\ ]$  como criterio principal de selección.
41.  $\text{CALCULA-}E(a)(\tau, t)$
42.  $k \leftarrow k + 1$
43.  $j \leftarrow j + 1$
44. **▷** Asigna destino a los vehículos suficientemente adelantados en la cola.
45.  $\text{ASIGNA-DESTINO}(v)$
46. **▷** Actualiza el congestionamiento del vértice.
47. *Calcula*  $-\iota(v)$
48.  $\text{CALCULA-}\rho(v)$
49. **▷** Selecciona los que efectivamente cruzaran.
50.  $\text{SeleccionaVeh}(v)$
51. **▷** Actualiza el congestionamiento del vértice.
52. *Calcula*  $-\iota(v)$

53.            CALCULA- $\rho(v)$
54.             $i \leftarrow i + 1$
55.  $\triangleright$  Para cada flecha  $a$ , actualiza los atributos.
56.       **for**  $i \leftarrow 1$  **to**  $\text{CARDINAL}(F(G)[*])$
57.  $\triangleright$  Respecto a cada usuario  $\tau$  actualiza:
58.             $\text{MONITOREO-M}(a)(\tau, t)$
59.             $\text{CALCULA-TR}(a)(t)$
60.             $MF[n][\text{ind}(a)][t] \leftarrow F[t]$
61.             $i \leftarrow i + 1$
62.         $t \leftarrow t + 1$
63. Completa- $MTR(t)$
64. Construye- $\widehat{d}_y^*(G)$

## 4.5. Métodos

En esta sección se muestran las descripciones y funcionamiento de los métodos de los objetos definidos en la sección anterior.

Las listas o arreglos son objetos predefinidos con sus propios métodos, entre los cuales se encuentra  $\widetilde{\text{AÑADE}}()$  y  $\text{CARDINAL}()$ . El método  $\text{CARDINAL}(Lista[])$  regresa el número de elementos en la lista. El método  $\widetilde{\text{AÑADE}}(elemento, Lista[*])$  añade el elemento *elemento* al final de la lista  $Lista[]$ . Ambos corren en tiempo constante.

Para evitar confuciones, en métodos que se refieran a un índice en particular, marcamos dicho índice colocando  $*$  en los corchetes correspondientes; de esta forma  $\text{CARDINAL}(NacimientosPorMes[][*])$  regresa el número de años para los que hay datos,  $\text{CARDINAL}(NacimientosPorMes[*][])$  = 12 los meses del año, y  $\text{CARDINAL}(NacimientosPorMes[][])$  el número total de datos en el arreglo. El método  $\widetilde{\text{AÑADE}}()$ , coloca el elemento al final de la lista, respecto al índice marcado con  $*$ .

#### 4.5.1. CONSTRUYE- $N(v)(G)$

Determina la exvecindad e invecindad<sup>5</sup> de  $v$  guardándola en las listas atributo  $N^+[]$  y  $N^-[]$ , respectivamente. Requiere como parámetro una gráfica  $G$ . El tiempo de ejecución es de orden  $\mathcal{O}(\|F(G)\|)$ .

1.  $\triangleright$  Variables locales:
2. variable  $i, j, k \in \mathbb{N}$
3.     variable  $a \in \mathbf{FLECHA}$
4.     variable  $v \in \mathbf{VÉRTICE}$
5.  $\triangleright$  Vacía las listas.
6.  $N^+[] \leftarrow \emptyset$
7.  $N^-[] \leftarrow \emptyset$
8.  $i \leftarrow 1 \triangleright$  Inicializa el índice de la lista  $N^+$
9.  $k \leftarrow 1 \triangleright$  Inicializa el índice de la lista  $N^-$
10.  $\triangleright$  Para cada flecha  $a$  en  $F(G)$ :
11. **for**  $j \leftarrow 1$  **to**  $\text{CARDINAL}(\mathbf{F}(G)[*])$
12.      $a \leftarrow F(G)[j]$
13.  $\triangleright$  Almacena en  $N^+$  ( $N^-$ ) un apuntador a la flecha  $a$ , si ésta tiene a  $v$  como vértice inicial (final).
14.     **if**  $v = \text{vini}(a)$
15.          $N^+[i] \leftarrow a$
16.          $i \leftarrow i + 1 \triangleright$  En cuyo caso avanza el índice de la lista.
17.     **if**  $v = \text{vfin}(a)$
18.          $N^-[k] \leftarrow a$
19.          $k \leftarrow k + 1 \triangleright$  En cuyo caso avanza el índice de la lista.
20.      $j \leftarrow j + 1$

---

<sup>5</sup>La noción de vecindad de un vértice sólo tiene sentido dentro de una gráfica, que en este procedimiento se incluye como parámetro. De hecho, el resultado del procedimiento son las vecindades del vértice  $v$  en la gráfica  $G$ , lo cual no se denota explícitamente. Ya que diferentes gráficas pueden utilizar el mismo vértice, los datos de  $N^+$  y  $N^-$  podrían no corresponder a ambas gráficas, sino a la última gráfica con la que se utilizó el método CONSTRUYE- $N$ .

### 4.5.2. CALCULA- $\rho(v)(t)$

Calcula, para  $v$ , el parámetro de congestión  $\rho(v)$  mediante la fórmula 3.6. Requiere de los atributos  $\mathfrak{I}_a, D_\lambda, \Upsilon$  y  $\mu$  actualizados. El tiempo de ejecución es de orden  $\mathcal{O}(\mathfrak{I})$ .

1.  $\triangleright$  Variables locales:
2. variable  $q, i, j \in \mathbb{N}$
3. variable  $a \in FLECHA$
4. variable  $\tau \in USUARIO$
5. variable  $\gamma \in VEHÍCULO$
6. variable  $Sum \in \mathbb{R}$
7.  $Sum \leftarrow 0$
8.  $\triangleright$  Para cada flecha  $a$  de la exvecindad de  $v$ :
9. **for**  $i \leftarrow 1$  **to**  $CARDINAL(N^+[*])$
10.      $a \leftarrow N^+[i]$
11.  $\triangleright$  Suma en la variable  $Sum$  el valor de la congestión de cada flecha de salida.
12.  $\triangleright$  Caso de indeterminación:
13.     **if**  $D_\lambda[i] = 0$
14.          $Sum \leftarrow Sum + 1$
15.     **else**
16.  $\triangleright$  Primero, calcula  $\lambda_\emptyset, \lambda_\zeta$  y  $\lambda_M$ .
17.      $\lambda_M(a) \leftarrow 0$
18.      $\lambda_\zeta(a) \leftarrow 0$
19.     **for**  $j \leftarrow 1$  **to**  $CARDINAL(\mathfrak{I}_a[*])$
20.          $\tau \leftarrow \mathfrak{I}_a[j]$
21.          $\gamma \leftarrow veh(\tau)$
22.          $\lambda_M(a) \leftarrow \lambda_M(a) + \varsigma(\gamma) \cdot \mu(a)$

23.  $\lambda_\zeta(a) \leftarrow \lambda_\zeta(a) + \varsigma(\gamma) \cdot \Upsilon(a)$
24.  $j \leftarrow j + 1$
25.  $\lambda_\emptyset(a) \leftarrow \lambda(a) \times \kappa(a) - \lambda_\zeta - \lambda_M$
26.  $\triangleright$  Calcula y suma el cociente del espacio de salida de  $a$  entre su demanda de espacio  $D_\lambda$
27.  $\triangleright$  para obtener una tasa de congestión. Si la demanda es menor que el
28.  $\triangleright$  espacio la tasa es 1.
29.  $Sum \leftarrow Sum + \text{mín}(1, \frac{\lambda_\emptyset(a)}{D_\lambda[ind(a)]})$
30.  $i \leftarrow i + 1$
31.  $\rho \leftarrow \frac{Sum}{i-1}$

#### 4.5.3. ASIGNA-DESTINO( $v$ )

Para todas las flechas en la invecindad del vértice  $v$  y todos los usuarios  $\tau$ , calcula de acuerdo con las fórmulas 3.13 y 3.14 el valor  $\zeta(a)[2][i] = ind(b)$ : la flecha destino de cada grupo  $i$  de vehículos en la cola de  $a$ . Requiere actualizados los atributos  $\iota$ ,  $E_\tau$ ,  $N^+$  y  $N^-$  en relación con la gráfica en cuestión. El tiempo de ejecución es de orden constante.

1.  $\triangleright$  Variables locales:
2. variable  $i, m, k, q, z, \bar{\tau}, \bar{b}, adel \in \mathbb{N}$
3. variable  $\tau \in \mathbf{USUARIO}$
4. variable  $a, b, c \in \mathbf{FLECHA}$
5.  $\triangleright$  Para cada flecha  $a$  en la invecindad de  $v$ .
6. **for**  $i \leftarrow 1$  **to**  $\text{CARDINAL}(N^-[*])$
7.  $a \leftarrow N^-(v)[i]$
8.  $\triangleright$  Para cada grupo de vehículos  $(m, \bar{b}, \bar{\tau})$  en  $\zeta(a)[*]$ :
9.  $k \leftarrow 1$
10.  $adel \leftarrow 0$
11. **while**  $\{adel \leq \iota(a) \text{ y } k \leq \text{CARDINAL}(\zeta(a)[*])\}$

12.  $(m, \bar{b}, \bar{\tau}) \leftarrow \zeta(a)[[k]$
13. ▷ Recupera el usuario  $\tau$  utilizando su índice.
14.  $\tau \leftarrow \mathfrak{I}(G)[\bar{\tau}]$
15. ▷ A los grupos de vehículos, que aún no tienen flecha destino asignada, les
16. ▷ asigna una si están suficientemente adelante en la cola.
17. **if**  $\bar{b} = \emptyset$
18. ▷ Establece  $b$  como la flecha que tendrá la mayor efectividad y  $r$  su parámetro
19. ▷ aleatorio de comportamiento, determinado por la función  $\Psi$ .
20.  $b \leftarrow N^+(v)[1]$
21.  $r \leftarrow \Psi$
22. ▷ Busca alguna otra flecha  $c$  mejor en términos de su *efectividad percibida*.
23.  $q \leftarrow 1$
24. **while**  $q < \text{CARDINAL-}N^+(v)[*]$
25.  $q \leftarrow q + 1$
26.  $c \leftarrow N^+(v)[q]$
27.  $s \leftarrow \Psi$
28. ▷ Si  $b$  es tal que su efectividad ponderada con el parámetro  $r$  es superada por la de otra flecha  $c$  y su parámetro  $s$ : Establece en  $b$  y  $r$  los valores de  $c$  y  $s$ .
29. **if**  $s \cdot E_\tau(c)[\bar{\tau}] > r \cdot E_\tau(b)[\bar{\tau}]$
30.  $b \leftarrow c$
31.  $r \leftarrow s$
32. ▷ Actualiza la Demanda de espacio de la flecha  $b$  elegida.
33.  $D_\lambda(b) \leftarrow D_\lambda(b) + \zeta(\text{veh}(\tau)) \times m$
34. ▷ Asigna  $b$ , al grupo de vehículos de  $\zeta$  en cuestión, como flecha destino.
35.  $\bar{b} \leftarrow \text{ind}(b)$
36.  $\zeta(a)[[k] \leftarrow (m, \bar{b}, \bar{\tau})$

37. ▷ Actualiza el número de vehículos adelantados en la fila.

38.  $adel \leftarrow adel + m$

39.  $k \leftarrow k + 1$

40.  $i \leftarrow i + 1$

#### 4.5.4. SELECCIONA-VEH( $v$ )()

Para todas las flechas  $b$  en la exvecindad de  $v$  y las flechas  $a$  de la invecindad, extrae los vehículos de  $\zeta(a)$  que cruzarán el vértice, colocandolos provisionalmente en  $I[ind(\tau)][ind(b)]$ , actualizando los valores  $D_\lambda(b)$  y  $\Upsilon(b)[ind(\tau)]$ . Corresponde al desahogo de cola de la subsección 3.3.1. Requiere de  $N^-$  respecto a la gráfica  $G$  en cuestión, lo mismo que  $\mathfrak{J}_a$ . El tiempo de ejecución es de orden constante.

1. ▷ Variables locales:
2. variable  $i, m, m^*, \bar{b}, \bar{\tau}, j, k, q, s, z \in \mathbb{N}$
3. variable  $a, b, c, p \in \mathbf{FLECHA}$
4. variable  $\tau \in \mathbf{USUARIO}$
5. ▷ Para cada flecha  $a$  en la invecindad de  $v$  y cada grupo de vehículos
6. ▷  $(m, \bar{b}, \bar{\tau})$  de la cola  $\zeta(a)$ :
7. **for**  $i \leftarrow 1$  **to**  $\text{CARDINAL}(N^-[*])$
8.  $a \leftarrow N^-[i]$
9.  $s \leftarrow 0$
10.  $j \leftarrow 0$
11. **for**  $k \leftarrow 1$  **to**  $\text{CARDINAL}(\zeta[*])$
12.  $(m, \bar{b}, \bar{\tau}) \leftarrow \zeta[k]$
13. ▷ Si el grupo de vehículos tiene una flecha asignada  $\bar{b} \neq \emptyset$
14. **if**  $\bar{b} \neq \emptyset$
15.  $b \leftarrow V(G)[\bar{b}]$

16.  $\tau \leftarrow \mathfrak{J}(G)[\bar{\tau}]$
17.  $q \leftarrow 0$
18.  $c \leftarrow \emptyset$
19.  $\triangleright$  Busca entre las flechas de la exvecindad la que corresponde a  $b$ .
20. **while**  $c \neq b$
21.  $q \leftarrow q + 1$
22.  $c \leftarrow N^+(v)[q]$
23. **if**  $b = c$
24.  $\triangleright$  Si todos los vehículos del grupo pueden cruzar hacia  $b$  durante el paso  $t$ ,
25.  $\triangleright$  los borra de la cola  $\zeta$  y aumenta en  $m$  la futura inyección de la
26.  $\triangleright$  flecha  $b$ .
27. **if**  $\begin{cases} m \cdot \varsigma(veh(\tau)) \leq \lambda_{\emptyset}(b) & \text{y} \\ j + m \leq \phi[ind(a)][ind(b)] \cdot \rho \cdot \kappa(a)[ind(b)] \cdot \pi \end{cases}$
28.  $BORRA(k, \zeta[[][*]])$
29.  $I[\bar{\tau}][\bar{b}] \leftarrow I[\bar{\tau}][\bar{b}] + m$
30.  $D_{\lambda}(b) \leftarrow D_{\lambda}(b) - \varsigma(veh(\tau)) \times m$
31.  $\Upsilon(a)[\bar{\tau}] \leftarrow \Upsilon(a)[\bar{\tau}] - m$
32.  $j \leftarrow j + m$
33.  $\mu(b)[\bar{\tau}] \leftarrow \mu(b)[\bar{\tau}] + m$
34.  $\triangleright$  Si sólo puede cruzar una parte del grupo, deja en la cola  $\zeta$  a la parte
35.  $\triangleright$  que no podrá cruzar y aumenta la inyección de  $b$  con la parte que sí
36.  $\triangleright$  podrá.
37. **if**  $j < \phi[ind(a)][ind(b)] \cdot \rho \cdot \kappa(a)[ind(b)] \cdot \pi < j + m$
38.  $m^* \leftarrow \min \left\{ \begin{array}{l} \phi[ind(a)][ind(b)] \cdot \rho \cdot \kappa(a)[ind(b)] \cdot \pi - j \\ \left\lfloor \frac{\lambda_{\emptyset}(b)}{\varsigma(veh(\tau))} \right\rfloor \end{array} \right\}$
39.  $\zeta[[][k]] \leftarrow (m - m^*, \bar{b}, \bar{\tau})$

40.  $I[\bar{\tau}][\bar{b}] \leftarrow I[\bar{\tau}][\bar{b}] + m^*$
41.  $D_\lambda(b) \leftarrow D_\lambda(b) - \varsigma(veh(\tau)) \times m^*$
42.  $\Upsilon(a)[\bar{\tau}] \leftarrow \Upsilon(a)[\bar{\tau}] - m^*$
43.  $\mu(b)[\bar{\tau}] \leftarrow \mu(b)[\bar{\tau}] + m^*$
44.  $k \leftarrow k + 1$
45.  $i \leftarrow i + 1$

#### 4.5.5. MONITOREO- $M(a)(\tau, t)$

Para la flecha  $a$  y cada usuario  $\tau$ : inyecta a  $M(a)[ind(\tau)]$  los vehículos que se incorporan a la flecha  $a$ , incluyendo aquellos que indica la función de inyección de la subsección 3.3.3; avanza los grupos de vehículos que están haciendo el recorrido y pasa a la cola  $\zeta(a)$  aquellos que lo terminaron. Adicionalmente, actualiza los atributos  $\lambda_M$ ,  $\lambda_\zeta$  y  $F$ . Corresponde en general al apartado 3.3.1 de seguimiento de vehículos. Requiere de la función de inyección  $I[ind(a)]$  actualizada, lo mismo que de  $TR(a)$ . El tiempo de ejecución es de orden  $\mathcal{O}(\mathfrak{I})$ .

1.  $\triangleright$  Variables locales:
2. variable  $i, j, m, \delta \in \mathbb{N}$
3. variable  $\tau \in \mathbf{USUARIO}$
4.  $\mu \leftarrow 0$
5.  $\Upsilon \leftarrow 0$
6.  $\triangleright$  Inyecta los vehículos provenientes de otras flechas o que inician recorrido.
7. **for**  $j \leftarrow 1$  **to**  $\mathbf{CARDINAL}(\mathfrak{I}_a[*])$
8.  $\tau \leftarrow \mathfrak{I}_a[j]$
9.  $\triangleright$  Decide si la flecha  $a$  es la flecha auxiliar del tipo  $\tau$  comparando sus índices ( $ind$  corresponde
10.  $\triangleright$  al de  $a$  pues el método que se define pertenece a **FLECHA**).
11. **if**  $ind = ind(h_\tau(\tau))$
12.  $\triangleright$  En caso afirmativo inyecta los vehículos que indica la demanda.
13.  $\mathbf{AÑADE}((dem(\tau)[t], TRC), M[ind(\tau)][*])$

14.  $\mu[ind(\tau)] \leftarrow \mu[ind(\tau)] + dem(\tau)[t]$
15. **else**
16.  $\triangleright$  En otro caso ( $a$  no es auxiliar) inyecta los vehículos provenientes de otras flechas.
17.  $\tilde{\text{AÑADE}}((I[ind(\tau)][ind], TRC), M[ind(\tau)][][*])$
18.  $\mu[ind(\tau)] \leftarrow \mu[ind(\tau)] + I[ind(\tau)][ind]$
19.  $I[ind(\tau)][ind] \leftarrow 0$
20.  $\triangleright$  Para cada grupo de vehículos  $(m, \delta)$  de  $M_\tau(a)$ .
21. **for**  $i \leftarrow 1$  **to**  $\text{CARDINAL}(M[ind(\tau)][][*])$
22.  $(m, \delta) \leftarrow M[ind(\tau)][][i]$
23.  $\triangleright$  Si el grupo todavía no llegó al extremo de la flecha, lo avanza.
24. **if**  $\delta > 0$
25.  $M[ind(\tau)][][i] \leftarrow (m, \delta - 1)$
26.  $\triangleright$  Si el grupo ya llegó al extremo, lo borra de  $M[ind(\tau)][][]$ .
27. **else if**  $\delta = 0$
28.  $\text{BORRA}(i, M[ind(\tau)][][*])$
29.  $\triangleright$  Si el vértice extremo no es el vértice destino del usuario, lo añade a la cola  $\zeta$ , si es el vértice destino no hace nada, dejando los vehículos fuera pues llegaron a su destino.
30. **if**  $v_{fin} \neq v_{dest}(\tau)$
31.  $\tilde{\text{AÑADE}}((m, \emptyset, ind(\tau)), \zeta[][*])$
32.  $\Upsilon[ind(\tau)] \leftarrow \Upsilon[ind(\tau)] + m$
33.  $\mu[ind(\tau)] \leftarrow \mu[ind(\tau)] - m$
34.  $i \leftarrow i + 1$
35.  $\mu \leftarrow \mu + \mu[ind(\tau)]$
36.  $\Upsilon \leftarrow \Upsilon + \Upsilon[ind(\tau)]$
37.  $j \leftarrow j + 1$
38.  $F \leftarrow \mu + \Upsilon$

#### 4.5.6. CALCULA- $TR(a)(t)$

Calcula, para la flecha  $a$ ,  $TRC$ ,  $\chi$  y  $TR$  de acuerdo a las fórmulas 3.3, 3.4 y 3.5, respectivamente. Obliga a la función  $TR$  a cumplir con la condición de fila del Apéndice A. Requiere de  $\iota$ , la intensidad de paso. El tiempo de ejecución es de orden constante.

1.  $\triangleright$  Variables locales:
2.  $vel \in \mathbb{R}^+$
3.  $\triangleright$  Cálculo de  $\chi$  mediante la fórmula 3.4.
4. **if**  $\iota = 0$
5.  $\chi \leftarrow \overline{\infty}$
6. **else**
7.  $\chi \leftarrow \frac{\gamma}{\iota}$
8.  $\triangleright$  Cálculo de  $TRC$  mediante la fórmula 3.3.
9.  $vel \leftarrow \frac{\lambda(a)}{\left(-10 + \sqrt{100 + (20 \times 3.28) \frac{\lambda_{\theta}(a,t)}{\mu(a,t)}}\right) \times 1.609}$
10.  $TRC \leftarrow \text{máx} \left\{ \frac{\lambda}{v}, vel \right\}$
11.  $TRC \leftarrow \text{mín} \{TRC, \varpi\}$
12.  $\triangleright$  Cálculo de  $TR$  con la condición de fila mediante las fórmulas 3.5 y A.1, respectivamente.
13. **if**  $TRC + \chi + 1 \geq TR$
14.  $TR \leftarrow TRC + \chi$
15. **else**
16.  $TR \leftarrow TR - 1$
17. **if**  $t + TR \leq \Lambda$
18.  $MTR[t + TR] \leftarrow TR$

#### 4.5.7. CALCULA- $E(a)(\tau, t)$

Calcula, para la flecha  $a$ , el valor de  $S_\tau(u, t)$  y  $S_\tau^*(v, t)$  para con ellos calcular  $E_\tau(a, t)$  mediante la fórmula 3.12. Requiere los valores actualizados de la distancia relativa  $\widehat{d}_y^\tau$ , el tiempo de recorrido  $TR$ , el espacio libre en las flechas de salida  $\lambda_\emptyset$  y su memoria  $M\lambda_\emptyset$ . El tiempo de ejecución es de orden constante.

1.  $\triangleright$  Variables locales:
2.  $sum \in \mathbb{R}$
3.  $i, m \in \mathbb{N}$
4. variable  $u, v \in \mathbf{VÉRTICE}$
5. variable  $b \in \mathbf{FLECHA}$
6. variable  $S, S^*, \widehat{TR}^\tau \in \mathbb{R}$
7.  $\triangleright$  Para los vértices inicial y final de la flecha:
8.  $u \leftarrow vini$
9.  $v \leftarrow vfin$
10.  $\triangleright$  calcula el valor de  $S_\tau(u, t)$  y  $S_\tau^*(v, t)$  mediante las fórmulas 3.10 y 3.11 respectivamente.
11.  $sum \leftarrow 0$
12. **for**  $i \leftarrow 1$  **to**  $CARDINAL(N^+(u)[*])$
13.  $b \leftarrow N^+(u)[i]$
14.  $sum \leftarrow sum + \frac{M\lambda_\emptyset(b) - \lambda_\emptyset(b)}{\lambda(b) \cdot \kappa(b)}$
15.  $i \leftarrow i + 1$
16.  $m \leftarrow CARDINAL(N^+(v)[*])$
17.  $S \leftarrow \widehat{d}_y^\tau(u)[ind(\tau)][t] \cdot \exp\left(\frac{sum}{m}\right)$
18.  $sum \leftarrow 0$
19. **for**  $i \leftarrow 1$  **to**  $CARDINAL(N^+(v)[*])$
20.  $b \leftarrow N^+(v)[i]$
21.  $sum \leftarrow sum + \frac{M\lambda_\emptyset(b) - \lambda_\emptyset(b)}{\lambda_\emptyset(b) \cdot \kappa(b)}$

22.  $i \leftarrow i + 1$
23.  $m \leftarrow \text{CARDINAL}(N^+(v)[*])$
24.  $S^* \leftarrow \widehat{d}_y^\tau(v)[\text{ind}(\tau)][t + TR] \cdot \exp\left(\frac{\text{sum}}{m}\right)$
25.  $\triangleright$  Almacena en el atributo  $E[]$  el valor de  $E_\tau(a, t)$ .
26.  $\widehat{TR}^\tau \leftarrow \frac{TR}{1+P_\tau}$
27.  $E[\text{ind}(\tau)] \leftarrow \frac{S-S^*}{\widehat{TR}^\tau}$

#### 4.5.8. CALCULA- $P_\tau(\tau)()$

Calcula, para la flecha  $a$  la preferencia total  $P_\tau$  mediante la fórmula 3.8. El tiempo de ejecución es de orden constante.

1.  $\triangleright$  Variables locales:
2. variable  $P, C \in \mathbb{R}$
3. variable  $c \in [0, 1]$
4. variable  $p \in [-1, 1]$
5.  $\triangleright$  Suma en  $P$  y  $C$  todos los valores de preferencias y características,
6.  $\triangleright$  respectivamente.
7.  $P \leftarrow 0$
8.  $C \leftarrow 0$
9.  $P_\tau \leftarrow 0$
10. **for**  $i \leftarrow 1$  **to**  $\text{CARDINAL}(\text{Car}[*])$
11.  $P \leftarrow P + \text{PREF}(\tau)[i]$
12.  $C \leftarrow C + \text{Car}[i]$
13.  $i \leftarrow i + 1$
14.  $\triangleright$  Para cada condición:
15.  $m \leftarrow \text{CARDINAL}(\text{Car}[*])$

16. **if**  $C < 1$
17.     **for**  $i \leftarrow 1$  **to**  $m$
18.          $p \leftarrow PREF(\tau)[i]$
19.          $c \leftarrow Car(\tau)[i]$
20.          $P_\tau \leftarrow P_\tau + p \times c$
21.          $i \leftarrow i + 1$
22. **if**  $C \geq 1$
23.     **for**  $i \leftarrow 1$  **to**  $m$
24.          $p \leftarrow PREF(\tau)[i]$
25.          $c \leftarrow Car(\tau)[i]$
26.          $P_\tau \leftarrow P_\tau + c \left( p + \frac{PI-P}{m-1} \right)$
27.          $i \leftarrow i + 1$
28.      $P_\tau \leftarrow P_\tau + \frac{P-PI}{m-1}$

## 4.6. Procedimientos

### 4.6.1. Borrar(*índice*, *Lista*[])

Borra el elemento No. *índice* de la lista *Lista*[], reacomodando los elementos restantes de forma que no queden huecos en la lista. El tiempo de ejecución es de orden  $\mathcal{O}(\|Lista\|)$ .

1.  $\triangleright$  Variables locales:
2.  $i \in \mathbb{N}$
3. **for**  $i \leftarrow \text{índice} + 1$  **to**  $CARDINAL(Lista[*])$
4.      $List[a[i-1]] \leftarrow List[a[i]]$
5.      $i \leftarrow i + 1$
6.  $List[CARDINAL(Lista[*])] \leftarrow \emptyset$

#### 4.6.2. **Buscar**(*elemento*, *Lista*[])

Busca el elemento *elemento* en la lista *Lista*[], Regresa el índice del elemento en la lista o  $\emptyset$  si el elemento no está en la lista. El tiempo de ejecución es de orden  $\mathcal{O}(\|Lista\|)$

1.  $\triangleright$  Variables locales:
2.  $i \in \mathbb{N}$
3. **while**  $elemento \neq Lista[i]$  **or**  $i > \text{CARDINAL}(Lista[])$
4.        $i \leftarrow i + 1$
5. **if**  $i > \text{CARDINAL}(Lista[])$
6.     **Return**  $\emptyset$
7. **else Return**  $i$

#### 4.6.3. **BuscarBorrar**(*elemento*, *Lista*[])

Borra el elemento *elemento* de la lista *Lista*[], acomodando los elementos restantes de forma que no queden huecos en la lista. El tiempo de ejecución es de orden  $\mathcal{O}(\|Lista\|)$ .

1.  $\triangleright$  Variables locales:
2.  $i \in \mathbb{N}$
3.  $i \leftarrow \text{Buscar}(elemento, Lista[])$
4. **Borrar**( $i, Lista[]$ )

#### 4.6.4. **UniónConj**(*Lista1*, *Lista2*, *ListaUnión*[])

Guarda en *ListaUnión* la unión de los conjuntos formados por los elementos de *Lista1* y *Lista2*. El tiempo de ejecución es de orden  $\mathcal{O}(\|Lista1\| \cdot \|Lista2\|)$

1.  $\triangleright$  Variables locales:

2.  $i, j, k \in \mathbb{N}$
3.  $k \leftarrow 1$
4. **for**  $i \leftarrow 1$  **to**  $\text{CARDINAL}(\text{Lista1}[*])$
5.      $j \leftarrow \text{Buscar}(\text{Lista1}[i], \text{Lista2}[])$
6.     **if**  $j \neq \emptyset$
7.          $\text{AÑADE}(\text{Lista1}[i], \text{ListaUnión}[k])$
8.          $k \leftarrow k + 1$
9.      $i \leftarrow i + 1$

#### 4.6.5. VolteaG( $G$ )

Regresa la **GRÁFICA**  $\tilde{G}$ , que se obtiene de  $G$  cambiando el sentido de todas las flechas. Intercambia, en consecuencia,  $N^+$  y  $N^-$ ,  $d_x^r$  y  $d_y^r$ . El tiempo de ejecución es de orden  $\mathcal{O}(\|V(G)\| + \|F(G)\|)$ .

1.  $\triangleright$  Variables locales:
2. variable  $i, d \in \mathbb{N}$
3. variable  $a, D[] \in \mathbf{FLECHA}$
4. variable  $v \in \mathbf{VÉRTICE}$
5. variable  $\tilde{G} \in \mathbf{GRÁFICA}$
6.  $\triangleright$  Añade a  $V(\tilde{G})$  los vértices de  $G$  cambiando sus exvecindades por
7.  $\triangleright$      invecindades y viceversa; lo mismo que  $d_x^r$  y  $d_y^r$ .
8.  $V(\tilde{G}) \leftarrow \emptyset$
9. **for**  $i \leftarrow 1$  **to**  $\text{CARDINAL}(V(G)[*])$
10.      $v \leftarrow V(G)[i]$
11.      $V(\tilde{G})[i] \leftarrow v$
12.      $d \leftarrow d_x^r(v)$

13.  $d_x^r(v) \leftarrow d_y^r(v)$
14.  $d_y^r(v) \leftarrow d$
15.  $D[] \leftarrow N^+(v)[*]$
16.  $N^+(v)[*] \leftarrow N^-(v)[*]$
17.  $N^-(v)[*] \leftarrow D[*]$
18.  $i \leftarrow i + 1$
19.  $\triangleright$  Añade a  $F(\tilde{G})$  las flechas de  $G$  intercambiando para cada flecha los vértices
20.  $\triangleright$  inicial y final.
21.  $F(\tilde{G}) \leftarrow \emptyset$
22. **for**  $i \leftarrow 1$  **to**  $\text{CARDINAL}(F(G)[*])$
23.  $F(G) \leftarrow a$
24.  $v \leftarrow \text{vfin}(a)$
25.  $\text{vfin}(a) \leftarrow \text{vini}(a)$
26.  $\text{vini}(a) \leftarrow v$
27.  $i \leftarrow i + 1$
28. **return**  $\tilde{G}$

#### 4.6.6. Calcula- $\iota(v)$

Para todas las flechas en la invecindad del vértice  $v$ , calcula el valor de  $\iota$  mediante la fórmula 3.7. Requiere los valores de  $N^+$  y  $N^-$ , en relación con la gráfica en cuestión, y de la congestión  $\rho(v)$ . El tiempo de ejecución es de orden constante.

1.  $\triangleright$  Variables locales:
2. variable  $i, j \in \mathbb{N}$
3. variable  $sum \in \mathbb{R}$
4. variable  $a, c \in \mathbf{FLECHA}$
5.  $\triangleright$  Para cada flecha  $a$  en la invecindad de  $v$ :

6. **for**  $i \leftarrow 1$  **to**  $\text{CARDINAL}(N^-[*])$
7.      $a \leftarrow N^-[i]$
8.  $\triangleright$      calcula en  $sum$  la suma de las intensidades de paso hacia todas las flechas  $c$  en la
9.  $\triangleright$      exvecindad de  $v$ .
10.      $sum \leftarrow 0$
11.     **for**  $j \leftarrow 1$  **to**  $\text{CARDINAL}(N^+[*])$
12.          $c \leftarrow N^+[j]$
13.          $sum \leftarrow sum + \phi[ind(a)][ind(c)] \cdot \kappa(a)[ind(c)]$
14.          $j \leftarrow j + 1$
15.      $\iota \leftarrow \lfloor 0.5 \cdot \rho \cdot \pi \cdot \kappa(a) \cdot sum \rfloor$
16.      $i \leftarrow i + 1$

#### 4.6.7. Completa-MTR

Completa, para todas las flechas en  $\tilde{G}$  y todos los tiempos en  $t \leq \Lambda$ , el valor de  $MTR(a)[t]$  y posteriormente voltea el parámetro de tiempo, de forma que empiece al final de la jornada y se acerque al inicio de ésta conforme avance. Es decir, se cumple  $\widetilde{MTR}(a)[t] = MTR(a)[\Lambda - t]$ , donde  $\widetilde{MTR}$  es (momentáneamente) la versión resultante del este procedimiento; al final del mismo asignamos los valores de  $\widetilde{MTR}$  a  $MTR$ , volteándola y completándola. El tiempo de ejecución es de orden  $\mathcal{O}(\Lambda \cdot \|F(G)\|)$ .

1.  $\triangleright$  Variables locales:
2.  $i, t, \widetilde{MTR}[] \in \mathbb{N}$
3.  $a \in \mathbf{FLECHA}$
4.  $\triangleright$  Para cada flecha  $a$  y tiempo  $t$ :
5. **for**  $i \leftarrow 1$  **to**  $\text{CARDINAL}(F(\tilde{G})[*])$
6.      $a \leftarrow F(\tilde{G})[i]$
7.     **for**  $t \leftarrow 1$  **to**  $\lceil \frac{\Lambda}{v} \rceil$

8.  $MTR[ind(a)][t] \leftarrow \frac{\lambda}{v}$
9.  $t \leftarrow t + 1$
10. **for**  $t \leftarrow \lceil \frac{\lambda}{v} \rceil$  **to**  $\Lambda$
11.  $\triangleright$  Si existen datos en la memoria para  $a$ :
12. **if**  $MTR(a)[t] \neq \emptyset$
13.  $\widetilde{MTR}[\Lambda - t] \leftarrow MTR(a)[t]$
14.  $\triangleright$  Si no existen datos en la memoria para  $a$  en el tiempo  $t$ , se fija en el tiempo que se le asignó en el paso anterior
15.  $\triangleright$   $(t - 1)$  y determina el valor  $MTR$  de acuerdo a la fórmula (3.16) de la sección 3.6.9.
16. **else if**  $MTR(a)[t] = \emptyset$
17.  $\widetilde{MTR}[\Lambda - t] \leftarrow MTR(a)[t - 1] + 1$
18.  $t \leftarrow t + 1$
19.  $i \leftarrow i + 1$
20.  $MTR(a)[] \leftarrow \widetilde{MTR}[]$

#### 4.6.8. Construye- $d_y^r(G)$

Aplica Dijkstra dinámico para cada tiempo inicial y vértice de origen de cada usuario. Voltea el parámetro de tiempo (ya que el procedimiento VolteaG no lo hace) y completa la distancia dinámica de cada vértice en los tiempos  $t$  para los que no quedó definida (ver A.2.3). El tiempo de ejecución es de orden  $\mathcal{O}(\|\mathfrak{J}\| \cdot \Lambda \cdot \|V(G)\| \cdot (\|V(G)\| + \Lambda))$ .

1.  $\triangleright$  Variables locales:
2. variable  $i, j, t, s, d[][] \in \mathbb{N}$
3. variable  $v \in \mathbf{VÉRTICE}$
4. variable  $\tilde{G} \in \mathbf{GRÁFICA}$
5.  $\triangleright$  Vacía la variable auxiliar de distancia.
6.  $d[][] \leftarrow \emptyset$

7. ▷ Voltea la gráfica para obtener la distancia al destino (Dijkstra es para obtener
8. ▷ distancias desde un origen).
9. VolteaG( $G, \tilde{G}$ )
10. ▷ Para cada tipo de usuario y tiempo de llegada, calcula la distancia en reversa desde el
11. ▷ vértice destino y la almacena en la variable  $d$  *volteando* el parámetro  $\tilde{t}$ .
12. **for**  $i \leftarrow 1$  **to** CARDINAL( $\mathfrak{J}(G)[*]$ )
13.      $\tau \leftarrow \mathfrak{J}(G)[i]$
14.     **for**  $t \leftarrow 1$  **to**  $\Lambda$
15.          $d[\ ][\Lambda - t] \leftarrow DJTdin(\tilde{G}, \tau, MTR()[ ], vdest(\tau), t)$
16.          $t \leftarrow t + 1$
17. ▷ Regresa G a su estado original.
18.     VolteaG( $\tilde{G}, G$ )
19. ▷ Construye  $\hat{d}_y^\tau$  a partir de la fórmula (A.5) de la sección A.2.3.
20.     **for**  $j \leftarrow 1$  **to** CARDINAL( $V(G)[*]$ )
21.          $v \leftarrow V(G)[j]$
22.         **for**  $t \leftarrow 1$  **to**  $\Lambda$
23.              $s \leftarrow d[ind(v)][d_y^\tau(v)[ind(\tau)]] [t]$
24.             **if**  $s < d_y^\tau(v)[ind(\tau)] [t - s]$
25.                  $d_y^\tau(v)[ind(\tau)] [t - s] \leftarrow s$
26.              $t \leftarrow t + 1$
27.          $j \leftarrow j + 1$
28. ▷ Se completa la distancia definiéndola para todo vértice en todo tiempo inicial.
29.     **for**  $j \leftarrow 1$  **to** CARDINAL( $V(G)[*]$ )
30.          $v \leftarrow V(G)[j]$
31. ▷ Para cada tiempo  $t$ , empezando por  $\Lambda$  y terminando en 1:
32.         **for**  $t \leftarrow \Lambda$  **to** 1

33. ▷ Si la distancia al destino no está definida para  $v$  en el tiempo  $t$ , se decreta a partir de la del tiempo
34. ▷  $t + 1$  (que en la recurrencia se define antes).
35. **if**  $d_y^r(v)[ind(\tau)][t] = \emptyset$
36.  $d_y^r(v)[ind(\tau)][t] \leftarrow d_y^r(v)[ind(\tau)][t + 1] + 1$
37.  $t \leftarrow t - 1$
38.  $j \leftarrow j + 1$
39.  $i \leftarrow i + 1$

#### 4.6.9. Construye- $\widehat{d}_y^r(G)$

Aplica Dijkstra escalonado sobre la gráfica relativa  $\mathfrak{E}(G, tr, \Lambda)$  para el vértice destino de cada usuario. Voltea el parámetro de tiempo (ya que el procedimiento VolteaG no lo hace) y completa la distancia relativa de cada vértice a los tiempos  $t$  para los que no quedó definida. El tiempo de ejecución es de orden  $\mathcal{O}(\|\mathfrak{J}\| \cdot \|V(G)\|^2 \cdot \Lambda^3)$ .

1. ▷ Variables locales:
2. variable  $i, j, t, s, d[\ ][\ ] \in \mathbb{N}$
3. variable  $v \in \mathbf{VÉRTICE}$
4. variable  $\tilde{G} \in \mathbf{GRÁFICA}$
5. ▷ Vacía la variable auxiliar de distancia.
6.  $d[\ ][\ ] \leftarrow \emptyset$
7. ▷ Voltea la gráfica para obtener la distancia al destino (Dijkstra es para obtener
8. ▷ distancias desde un origen).
9. VolteaG( $G, \tilde{G}$ )
10. ▷ Para cada tipo de usuario y tiempo de llegada, calcula la distancia en reversa desde el
11. ▷ vértice destino y la almacena en la variable  $d$  volteando el parámetro  $\tilde{t}$ .
12. **for**  $i \leftarrow 1$  **to** CARDINAL( $\mathfrak{J}(G)[*]$ )

13.  $\tau \leftarrow \mathfrak{I}(G)[i]$
14.  $\triangleright$  Calcula el tiempo relativo  $\widehat{TR}$  que utiliza Dijkstra.
15. **for**  $j \leftarrow 1$  **to**  $\text{CARDINAL}(F(G)[*])$
16.  $a \leftarrow F(G)[j]$
17. **for**  $s \leftarrow 1$  **to**  $\Lambda$
18. **if**  $1 + P_\tau(a)[\text{ind}(\tau)] \leq 0$
19.  $\widehat{TR}[\text{ind}(\tau)][s] \leftarrow \infty$
20. **else**
21.  $\widehat{TR}[\text{ind}(\tau)][s] \leftarrow \frac{MTR(a)[s]}{1 + P_\tau(a)[\text{ind}(\tau)]}$
22.  $s \leftarrow s + 1$
23.  $j \leftarrow j + 1$
24. **for**  $t \leftarrow 1$  **to**  $\Lambda$
25.  $d[\Lambda - s] \leftarrow DJTesc(\widetilde{G}, \widehat{TR}[\text{ind}(\tau)], \tau, MTR(), vdest(\tau), t)$
26.  $t \leftarrow t + 1$
27.  $\triangleright$  Regresa  $G$  a su estado original.
28.  $\text{VolteaG}(\widetilde{G}, G)$
29.  $\triangleright$  Construye  $\widehat{d}_y^\tau$  a partir de la fórmula (A.5) de la sección A.2.3.
30. **for**  $j \leftarrow 1$  **to**  $\text{CARDINAL}(V(G)[*])$
31.  $v \leftarrow V(G)[j]$
32. **for**  $t \leftarrow 1$  **to**  $\Lambda$
33.  $s \leftarrow d[\text{ind}(v)][d_y^\tau(v)[\text{ind}(\tau)][t]$
34. **if**  $s < d_y^\tau(v)[\text{ind}(\tau)][t - s]$
35.  $d_y^\tau(v)[\text{ind}(\tau)][t - s] \leftarrow s$
36.  $t \leftarrow t + 1$
37.  $j \leftarrow j + 1$
38.  $\triangleright$  Se completa la distancia definiéndola para todo vértice en todo tiempo inicial.

39.     **for**  $j \leftarrow 1$  **to**  $\text{CARDINAL}(V(G)[*])$
40.              $v \leftarrow V(G)[j]$
41.  $\triangleright$  Para cada tiempo  $t$ , empezando por  $\Lambda$  y terminando en 1:
42.             **for**  $t \leftarrow \Lambda$  **to** 1
43.  $\triangleright$  Si la distancia al destino no está definida para  $v$  en el tiempo  $t$ , se decreta a partir de la del tiempo
44.  $\triangleright$       $t + 1$  (que en la recurrencia se define antes).
45.             **if**  $d_y^\tau(v)[\text{ind}(\tau)][t] = \emptyset$
46.                      $d_y^\tau(v)[\text{ind}(\tau)][t] \leftarrow d_y^\tau(v)[\text{ind}(\tau)][t + 1] + 1$
47.              $t \leftarrow t - 1$
48.              $j \leftarrow j + 1$
49.      $i \leftarrow i + 1$

#### 4.6.10. Construye $G_\gamma(G, \gamma)$

Construye la gráfica  $G_\gamma$  que, como se describe en 3.1.2, se obtiene de  $G$  tomando las flechas que están en el conjunto de vías adicionales del vehículo  $\gamma$ , o cuyo tipo de vía figure entre las vías aptas del mismo. El procedimiento requiere del método CONSTRUYE-N. El tiempo de ejecución es de orden  $\mathcal{O}(\|V(G)\| \cdot \|F(G)\|)$ .

1.  $\triangleright$  Variables locales:
2. variable  $i \in \mathbb{N}$
3. variable  $v \in \mathbf{VÉRTICE}$
4. variables  $G_\gamma \in \mathbf{GRÁFICA}$
5.  $\triangleright$  Para cada vértice  $v$  de  $V(G)$ :
6. **for**  $i \leftarrow 1$  **to**  $\text{CARDINAL}(V(G)[*])$
7.      $v \leftarrow V(G)[i]$
8.     CONSTRUYE-N( $v$ )( $G$ )

9. ▷ Inicializa  $e(v) = 0$  como indicador de que el vértice no tiene ninguna.
10. ▷     flecha en  $G_\gamma$ .
11.      $e(v) \leftarrow 0$
12.      $i \leftarrow i + 1$
13. ▷ Para cada vértice  $v$  de  $V(G)$ :
14. **for**  $i \leftarrow 1$  **to**  $\text{CARDINAL}(V(G)[*])$
15.     **for**  $j \leftarrow 1$  **to**  $\text{CARDINAL}(N^+(v)[*])$
16. ▷ Para cada flecha  $a$  de la exvecindad de  $v$ :
17.      $a \leftarrow N^+(v)[j]$
18.      $w \leftarrow \text{fin}(a)$
19. ▷ Determina si el tipo de vía de  $a$  es una vía apta o adicional del vehículo  $\gamma$ .
20.     **if**  $\{ \text{BUSCAR}(\vartheta(a), \theta(\gamma)[*]) \text{ or } \text{BUSCAR}(a, \theta^*(\gamma)[*]) \}$
21. ▷ En caso afirmativo, añade la flecha  $a$  como flecha de  $G_\gamma$ . Si no han sido añadidos antes (Si  $e(v) = 0$ ,
22. ▷      $e(w) = 0$ ), añade  $v$  y  $w$  como vértices de  $G_\gamma$  y establece  $e(v) = 1$ ,  $e(w) = 1$  indicando que los
23. ▷     vértices fueron añadidos.
24.      $\text{AÑADE}(a, F(G_\gamma)[*])$
25.     **if**  $e(v) = 0$
26.      $\text{AÑADE}(v, V(G_\gamma)[*])$
27.      $e(v) \leftarrow 1$
28.     **if**  $e(w) = 0$
29.      $\text{AÑADE}(w, V(G_\gamma)[*])$
30.      $e(w) \leftarrow 1$
31.      $j \leftarrow j + 1$
32.      $i \leftarrow i + 1$
33. ▷ Regresa la gráfica  $G_\gamma$ .
34. **return**  $G_\gamma$

#### 4.6.11. Dijkstra, DJT( $G, x, f$ )

Inicia con una gráfica *correcta*  $G$ , una función de costos  $f: F(G) \rightarrow \mathbb{R}^+$  constante en el tiempo, sobre las flechas de dicha gráfica, y  $x$  como vértice raíz. Regresa el valor  $d_x^r(v) \forall v \in V(G)$ , la distancia del vértice  $x$  al  $v$ . Requiere de los valores  $N^+$  respecto a la gráfica en cuestión. El tiempo de ejecución es de orden  $\mathcal{O}(\|V(G)\|^2)$ .

1.  $\triangleright$  Variables locales:
2. variable  $i, j \in \mathbb{N}$
3. variable  $Prov[]$ ,  $z, v \in \mathbf{VÉRTICE}$
4. variable  $a \in \mathbf{FLECHA}$
5.  $\triangleright$  Inicializa todos los vértices como provisionales, etiquetándolos  $e = 0$
6.  $\triangleright$  e inicializa sus distancias  $d_x^r = \infty$ .
7.  $i \leftarrow 1$
8. **for**  $j \leftarrow 1$  **to**  $\text{CARDINAL}(\mathbf{V}(G)[*])$
9.  $v \leftarrow V(G)[j]$
10.  $d_x^r(v) \leftarrow \infty$
11.  $e(v) \leftarrow 0$
12. **if**  $v \neq x$
13.  $Prov[i] \leftarrow v$
14.  $i \leftarrow i + 1$
15.  $j \leftarrow j + 1$
16.  $\triangleright$  Almacena los vértice provisionales en  $Prov[]$  estableciendo el origen
17.  $\triangleright$   $x$  como el único vértice permanente.
18.  $e(x) \leftarrow 1$
19.  $d_x^r(x) \leftarrow 0$
20.  $\triangleright$  Establece el nuevo vértice permanente  $z$  en el vértice  $x$ .
21.  $z \leftarrow x$

22. ▷ Mientras haya vértices provisionales alcanzables desde los permanentes:
23. **while** { $CARDINAL(Prov[]) > 0$  **and**  $d(z) \neq \infty$ }
24. ▷ Para cada flecha  $a$  de la exvecindad del nuevo vértice permanente  $z$ , actualiza la distancia mínima
25. ▷ provisional  $d_x^r(v)$  del vértice  $v$  que está al final de  $a$ .
26.     **for**  $j \leftarrow 1$  **to**  $CARDINAL(N^+(z)[*])$
27.          $a \leftarrow N^+(z)[j]$
28.          $v \leftarrow vfin(a)$
29. ▷ reestabeciéndola si mejora con una ruta que pase por  $z$ .
30.         **if**  $d_x^r(v) > d_x^r(z) + f(a)$
31.              $d_x^r(v) \leftarrow d_x^r(z) + f(a)$
32.          $j \leftarrow j + 1$
33. ▷ Escoge como nuevo vértice permanente  $z$  aquel de los provisionales
34. ▷ cuya distancia provisional  $d_x^r$  sea la menor.
35.      $z \leftarrow Prov[1]$
36.     **for**  $j \leftarrow 2$  **to**  $CARDINAL(Prov[*])$
37.          $v \leftarrow Prov[j]$
38.         **if**  $d_x^r(z) > d_x^r(v)$
39.              $z \leftarrow v$
40.          $j \leftarrow j + 1$
41. ▷ Establece el nuevo  $z$  como permanente y lo borra de los provisionales.
42.      $e(z) \leftarrow 1$
43.     *BuscarBorrar*( $z, Prov[*]$ )

#### 4.6.12. Dijkstra dinámico, DJTdin( $G, \tau, f, y, t_0$ )

Inicia con una gráfica *correcta*  $G$ , una función de costos  $f(a, t): F(G) \times [0, \Lambda) \rightarrow \mathbb{R}^+$ , en las flechas de dicha gráfica,  $y$  como vértice raíz y  $t_0$  como tiempo de inicio. Regresa la lista de valores  $d_y^\tau(v)[ind(\tau)][t_0] \forall v \in V(G)$ , la distancia dinámica del vértice  $y$  al  $v$  partiendo en el tiempo  $t_0$ . Requiere de los valores  $N^+$  relativos a la gráfica en cuestión. El tiempo de ejecución es de orden  $\mathcal{O}(\|V(G)\|^2)$ .

1.  $\triangleright$  Variables locales:
2. variable  $d[], i, j \in \mathbb{N}$
3. variable  $Prov[], z, v, y \in \mathbf{VÉRTICE}$
4. variable  $a \in \mathbf{FLECHA}$
5.  $y \leftarrow vdest(\tau)$
6.  $\triangleright$  Inicializa todos los vértices como provisionales, etiquetándolos  $e = 0$
7.  $\triangleright$  e inicializa sus distancias  $d[] = \infty$ .
8.  $i \leftarrow 1$
9. **for**  $j \leftarrow 1$  **to**  $CARDINAL(\mathbf{V}(G)[*])$
10.  $v \leftarrow V(G)[j]$
11.  $d[ind(v)] \leftarrow \infty$
12.  $e(v) \leftarrow 0$
13.  $\triangleright$  Almacena los vértices provisionales en  $Prov[]$  estableciendo el origen
14.  $\triangleright$   $y$  como el único vértice permanente.
15. **if**  $v \neq y$
16.  $Prov[i] \leftarrow v$
17.  $i \leftarrow i + 1$
18.  $j \leftarrow j + 1$
19.  $e(y) \leftarrow 1$
20.  $d[ind(y)] \leftarrow 0$

21. ▷ Establece el nuevo vértice permanente  $z$  en el vértice  $y$ .
22.  $z \leftarrow y$
23. ▷ Mientras haya vértices provisionales alcanzables desde los permanentes:
24. **while**  $\{ \text{CARDINAL}(\text{Prov}[]) > 0 \text{ and } d[\text{ind}(z)] \neq \infty \}$
25. ▷ Para cada vertice  $v$  de la exvecindad (en vértices) del nuevo vértice
26. ▷ permanente  $z$ , actualiza la distancia mínima provisional  $d[\text{ind}(v)]$ .
27.     **for**  $j \leftarrow 1$  **to**  $\text{CARDINAL}(N^+(z)[*])$
28.          $a \leftarrow N^+(z)[j]$
29.          $v \leftarrow \text{vfin}(a)$
30. ▷ reestabeciéndola si mejora llegando directamente desde  $z$ .
31.         **if**  $d[\text{ind}(v)] > d[\text{ind}(z)] + f(a, d[\text{ind}(z)] + t_0)$
32.              $d[\text{ind}(v)] \leftarrow d[\text{ind}(z)] + f(a, d[\text{ind}(z)] + t_0)$
33.          $j \leftarrow j + 1$
34. ▷ Escoge como nuevo vértice permanente  $z$  aquel de los provisionales
35. ▷ cuya distancia provisional  $d[\text{ind}()]$  sea la menor.
36.      $z \leftarrow \text{Prov}[1]$
37.     **for**  $j \leftarrow 2$  **to**  $\text{CARDINAL}(\text{Prov}[*])$
38.          $v \leftarrow \text{Prov}[j]$
39.         **if**  $d[\text{ind}(z)] > d[\text{ind}(v)]$
40.              $z \leftarrow v$
41.          $j \leftarrow j + 1$
42. ▷ Establece el nuevo  $z$  como permanente y lo borra de los provisionales.
43.      $e(z) \leftarrow 1$
44.      $\text{BuscarBorrar}(z, \text{Prov}[*])$
45. **return**  $d[]$

#### 4.6.13. Dijkstra escalonado, DJTdin( $G, \tau, c, tr, y, t_0$ )

Inicia con una gráfica *correcta*  $G$ , una función de costos  $f(a, t): F(G) \times [0, \Lambda] \rightarrow \mathbb{R}^+$ , en las flechas de dicha gráfica,  $y$  como vértice raíz y  $t_0$  como tiempo de inicio. Utiliza implícitamente, apoyado en variables locales, la gráfica escalonada  $\mathfrak{E}(G, f, \Lambda)$  de modo que los vértices y flechas de esta gráfica se tratarán como parejas ordenadas  $(v, t)$  y  $(a, t)$ . Llamaremos pareja provisional a la correspondiente a un vértice provisional en  $\mathfrak{E}(G, f, \Lambda)$ , las otras definiciones para vértices y aristas se tratarán de forma análoga. Regresa la distancia relativa  $\widehat{d}_y^{\tau}(v)[ind(\tau)][t_0] \forall v \in V(G)$  del vértice  $y$  al  $v$  partiendo en el tiempo  $t_0$ . Requiere de los valores  $N^+$  relativos a la gráfica  $G$ . El tiempo de ejecución es de orden  $\mathcal{O}(\|V(G) \times \Lambda\|^2)$ .

1.  $\triangleright$  Variables locales:
2. variable  $c[\ ][\ ], d[\ ][\ ], i, j, s \in \mathbb{N}$
3. variable  $z, v, y \in \mathbf{VÉRTICE}$
4. variable  $a \in \mathbf{FLECHA}$
5. variable  $Prov[\ ], e[\ ][\ ] \in \{0, 1\}$
6.  $y \leftarrow vdest(\tau)$
7.  $\triangleright$  Inicializa todos los vértices como provisionales, etiquetándolos  $e[\ ][\ ] = 0$
8.  $\triangleright$   $e$  inicializa sus distancias  $d[\ ][\ ] = \infty$  y costos  $c[\ ][\ ] = \infty$ .
9.  $i \leftarrow 1$
10. **for**  $j \leftarrow 1$  **to** CARDINAL( $V(G)[*]$ )
11.  $v \leftarrow V(G)[j]$
12. **for**  $s \leftarrow t$  **to**  $\Lambda$
13.  $d[ind(v)][s] \leftarrow \infty$
14.  $e[ind(v)][s] \leftarrow 0$
15.  $s \leftarrow s + 1$
16.  $j \leftarrow j + 1$
17.  $e[ind(y)][t] \leftarrow 1$
18.  $d[ind(y)][t] \leftarrow 0$
19.  $\triangleright$  Establece la cardinalidad  $Prov$  de los vértices provisionales: todos excepto el origen  $y$ .

20.  $Prov \leftarrow (\text{CARDINAL}(V(G)[*]) \times (\Lambda - t + 1) - 1$
21.  $\triangleright$  Establece el nuevo vértice permanente  $(z, s_z)$  en el vértice  $(y, t)$ .
22.  $(z, s_z) \leftarrow (y, t)$
23.  $\triangleright$  Mientras haya vértices provisionales alcanzables desde los permanentes:
24. **while**  $\{Prov > 0 \text{ and } d[ind(z)][s_z] \neq \infty\}$
25.  $\triangleright$  Para cada vertice  $(v, s)$  de la exvecindad (en vértices) del nuevo vértice
26.  $\triangleright$  permanente  $(z, s_z)$ , actualiza la distancia mínima provisional  $d[ind(v)][s]$ .
27.     **for**  $j \leftarrow 1$  **to**  $\text{CARDINAL}(N^+(z)[*])$
28.          $a \leftarrow N^+(z)[j]$
29.          $v \leftarrow vfin(a)$
30.  $\triangleright$  reestabeciéndola si mejora llegando directamente desde  $(z, s_z)$ .
31.         **if**  $d[ind(v)][s_z + tr(a, s_z)] > d[ind(z)][s_z] + c(a, s_z)$
32.              $d[ind(v)][s_z + tr(a, s_z)] \leftarrow d[ind(z)][s_z] + c(a, s_z)$
33.          $j \leftarrow j + 1$
34.  $\triangleright$  Escoge como nuevo vértice permanente  $(z, s_z)$  aquel de los provisionales
35.  $\triangleright$  cuya distancia provisional  $d[\ ][\ ]$  sea la menor.
36.      $j \leftarrow 1$
37.     **while**  $j \leq \text{CARDINAL}(V(G)[*])$  **and**  $e[ind(v)][s] == 1$
38.          $s \leftarrow t$
39.          $j \leftarrow j + 1$
40.     **while**  $s \leq \Lambda$  **and**  $e[ind(v)][s] == 1$
41.          $v \leftarrow V(G)[j]$
42.          $s \leftarrow s + 1$
43.      $(z, s_z) \leftarrow (v, s)$
44.     **for**  $j \leftarrow j$  **to**  $\text{CARDINAL}(V(G)[*])$
45.         **for**  $s \leftarrow s + 1$  **to**  $\Lambda$

46.  $v \leftarrow V(G)[j]$
47. **if**  $e[ind(v)][s] == 0$  **and**  $d[ind(z)][s_z] > d[ind(v)][s]$
48.  $(z, s_z) \leftarrow (v, s)$
49.  $j \leftarrow j + 1$
50.  $\triangleright$  Establece el nuevo  $z$  como permanente y lo borra de los provisionales.
51.  $e[ind(z)][s_z] \leftarrow 1$
52.  $Prov \leftarrow Prov - 1$
53. **return**  $d[\ ]$

#### 4.6.14. EvalConv( $MF[\ ][\ ][\ ]$ , $N, \epsilon, G, n$ )

Determina, a partir de las últimas  $N$  jornadas, si la última jornada ( $n$ ), es  $\epsilon$  estable, es decir, si en las últimas  $N$  jornadas a variado en más de  $\epsilon$  en el sentido de la fórmula 3.15. Regresa **TRUE** si se alcanzó la estabilidad y **FALSE** si no se alcanzó o no puede determinarse.

1.  $\triangleright$  Variables locales:
2. variable  $Sum, D \in \mathbb{R}^+$
3. variable  $i, t, m \in \mathbb{N}$
4. variable  $a \in \mathbf{FLECHA}$
5.  $Sum \leftarrow 0$
6.  $m \leftarrow \text{CARDINAL}(F(G)[*])$
7.  $\triangleright$  Calcula, si tiene sentido, la diferencia con la jornada anterior, en términos de la fórmula 3.18.
8. **if**  $1 < n$
9. **for**  $i \leftarrow 1$  **to**  $m$
10.  $a \leftarrow F(G)[i]$
11. **for**  $t \leftarrow 1$  **to**  $\Lambda$
12.  $D \leftarrow \frac{MF[n][ind(a)][t] - MF[n-1][ind(a)][t]}{\max\{1, MF[n-1][ind(a)][t]\}}$

13.  $t \leftarrow t + 1$
14.  $i \leftarrow i + 1$
15.  $\triangleright$  Si no se tiene la muestra completa, almacena la diferencia obtenida e indica que no puede determinarse la estabilidad (regresando el valor **False**).
16. **if**  $n < N$
17.  $\Delta[n - 1] \leftarrow D$
18. **return FALSE**
19.  $\triangleright$  Si se tiene una muestra completa, calcula la estabilidad de acuerdo a la fórmula 3.15.
20. **else**
21.  $\triangleright$  Si ya se tenía una muestra completa ( $n > N$ ), actualiza la lista  $\Delta$  borrando el primer elemento, que ya no es necesario, y recorriendo los otros valores de la lista. Suma en  $Sum$  las diferencias que prevalecen en la lista.
22. **if**  $n > N$
23. **for**  $i \leftarrow 1$  **to**  $N - 2$
24.  $\Delta[i] \leftarrow \Delta[i + 1]$
25.  $Sum \leftarrow Sum + \Delta[i]$
26.  $i \leftarrow i + 1$
27.  $\triangleright$  Incorpora la nueva diferencia a la lista y la suma en  $Sum$ .
28.  $\Delta[N - 1] \leftarrow D$
29.  $Sum \leftarrow Sum + \Delta[N - 1]$
30.  $\triangleright$  Si  $n = N$  suma en  $Sum$  las diferencias.
31. **else if**  $n = N$
32. **for**  $i \leftarrow 1$  **to**  $N - 1$
33.  $\Delta[i] \leftarrow \Delta[i + 1]$
34.  $Sum \leftarrow Sum + \Delta[i]$
35.  $i \leftarrow i + 1$
36.  $\triangleright$  Calcula en  $Sum$  la estabilidad.

37.  $Sum \leftarrow \frac{Sum}{m \times \Lambda \times N}$
38.  $\triangleright$  Si la estabilidad es menor que  $\epsilon$  regresa **Verdadero**, en caso contrario regresa **Falso**.
39. **if**  $Sum < \epsilon$  **return TRUE**
40. **else return FALSE**

## 4.7. Procedimientos especiales

### 4.7.1. NivelDeServicio( $G, MTR^*(\cdot), TAB[\cdot][\cdot], \eta$ )

Tomando  $NS(a)$  como un atributo extra de **FLECHA**, calcula el nivel de servicio  $NS(a)$  de las flechas de la gráfica  $G$  utilizando la Memoria de tiempo de recorrido (\*) estándar<sup>6</sup> definida como  $MTR^*(a)[t] = TR(a)[t]$ . Como dato extra de entrada utiliza los datos del Cuadro 3.1 en la sección 3.5.1, mismos que incorpora en dentro de la lista doble  $TAB[\cdot][\cdot]$ . Definimos  $TAB$  de modo que:

$$\left. \begin{array}{l} TAB[i][7] = \text{Rango } i \text{ de velocidad libre} \\ TAB[i][k] = \text{Valor } i \times k \text{ de la TABLA ??} \end{array} \right\} 1 \leq i \leq 3, 1 \leq k \leq 6$$

Donde  $A \sim 1$ ,  $B \sim 2$ ,  $C \sim 3$ ,  $D \sim 4$ ,  $E \sim 5$  y  $F \sim 6$ . Finalmente, como criterio de cercanía a las velocidades libres marcadas por el manual utilizamos el *radio de cercanía*  $\eta$ . El tiempo de ejecución es de orden  $\mathcal{O}(\|F(G)\| \cdot \Lambda)$

1.  $\triangleright$  Variables locales:
2. variable  $i, j, k, t, s, p \in \mathbb{N}$
3. variable  $VMR, VMR^* \in \mathbb{R}^+$
4. variable  $a \in \mathbf{FLECHA}$
5.  $\triangleright$  Suponiendo que las unidades de  $\pi$  son segundos, calcula el número mínimo de pasos necesarios para cubrir 15 min.
6.  $p \leftarrow \lceil \frac{15 \times 60}{\pi} \rceil$
7. **for**  $j \leftarrow 1$  **to**  $CARDINAL(F(G)[*])$
8.  $a \leftarrow F(G)[j]$

---

<sup>6</sup>Esta memoria es un dato de procedimiento extra que no se calcula en el algoritmo descrito anteriormente; sin embargo se puede incorporar fácilmente.

```

9.    $VMR^* \leftarrow 0$ 
10.  for  $t \leftarrow 1$  to  $\Lambda$ 
11.       $VMR \leftarrow 0$ 
12.      for  $s \leftarrow 0$  to  $p - 1$ 
13.           $VMR \leftarrow VMR + TR(a, t + s)$ 
14.           $s \leftarrow s + 1$ 
15.      if  $VMR < VMR^*$ 
16.           $VMR^* \leftarrow VMR$ 
17.       $t \leftarrow t + 1$ 
18.   $i \leftarrow 1$ 
19.  while  $|v(a) - TAB[i][7]| > \eta$  and  $i < 4$ 
20.       $i \leftarrow i + 1$ 
21.  if  $i < 4$ 
22.       $k \leftarrow 1$ 
23.      while  $VMR^*(a) < TAB[i][k]$ 
24.           $k \leftarrow k + 1$ 
25.       $NS(a) \leftarrow k$ 
26.  if  $i \geq 4$ 
27.       $NS(a) \leftarrow \emptyset$ 
28.   $j \leftarrow j + 1$ 

```

#### 4.7.2. Energía Y contaminantes

Calcula para todas las flechas y tiempos, en los atributos extras  $EC(a)[t]$  y  $CE(a)[t]$ , la emisión de un contaminante dado y el gasto de combustible, por los vehículos que la cruzan durante una jornada. Calcula además en  $EC^*(a)$  y  $CE^*(a)$  la emisión y consumo de la cada flecha  $a$  para la jornada completa. Finalmente obtiene  $ECR$  y  $CER$ , la emisión y consumo de la red completa durante todas la jornada.

Requiere de los factores emisión y consumo  $F_E[\text{flecha}]$  y  $F_C[\text{flecha}]$  por flecha, modelo y tipo de vehículo<sup>7</sup>, y de las proporciones por modelo de cada tipo de vehículo  $P[m][\tau]$ . Utiliza una memoria extra  $Mw_{\tau}^{-}(a)[t]$  que indica el número de usuarios de tipo  $\tau$  que salieron de la flecha  $a$  en el tiempo  $t$ .

El tiempo de ejecución es de orden  $\mathcal{O}(\|F(G)\| \cdot \Lambda)$ .

1.  $\triangleright$  Variables locales:
2. variable  $j, k, m \in \mathbb{N}$
3. variable  $ECR, CER \in \mathbb{R}^+$
4. variable  $a \in \mathbf{FLECHA}$
5. variable  $\tau \in \mathbf{USUARIO}$
6.  $ECR \leftarrow 0$
7.  $\triangleright$  Para cada flecha:
8. **for**  $j \leftarrow 1$  **to**  $\text{CARDINAL}(F(G)[*])$
9.      $a \leftarrow F(G)[j]$
10.      $EC^*(a) \leftarrow 0$
11.      $CE^*(a) \leftarrow 0$
12.  $\triangleright$  Calcula para cada  $t$  el valor de  $EC(a)[t]$  y  $CE(a)[t]$ :
13.     **for**  $t \leftarrow 1$  **to**  $\Lambda$
14.          $EC(a)[t] \leftarrow 0$
15.          $CE(a)[t] \leftarrow 0$
16.     **for**  $k \leftarrow 1$  **to**  $\text{CARDINAL}(\mathfrak{J}_a[*])$
17.          $\tau \leftarrow \mathfrak{J}_a[k]$
18.         **for**  $m \leftarrow 1$  **to**  $\text{CARDINAL}(MOD[*])$
19.  $\triangleright$  Calcula la emisión y consumo de los vehiculos con tipo, modelo dados que terminaron de recorrer la flecha  $a$  en el tiempo  $t$ .

<sup>7</sup>Muy probablemente dos tipos de vehículos  $\tau, \tau^* \in \mathfrak{J}$ , correspondan a un mismo tipo respecto los factores de emisión y consumo, por lo que es necesario desglosar estos factores de manera que los tipos se adapten a los del conjunto  $\mathfrak{J}$ .

20.  $Mw_{\tau}^{-}(a)[t] \leftarrow EC(a)[t] \leftarrow EC(a)[t] + F_C[ind(veh(\tau))][m][ind(a)] \times P[m][ind(\tau)] \times \lambda(a) \times$
21.  $Mw_{\tau}^{-}(a)[t] \leftarrow CE(a)[t] \leftarrow CE(a)[t] + F_C[ind(veh(\tau))][m][ind(a)] \times P[m][ind(\tau)] \times \lambda(a) \times$
22.  $m \leftarrow m + 1$
23.  $k \leftarrow k + 1$
24.  $EC^*(a) \leftarrow EC^*(a) + EC(a)[t]$
25.  $CE^*(a) \leftarrow CE^*(a) + CE(a)[t]$
26.  $t \leftarrow t + 1$
27.  $ECR \leftarrow ECR(a) + EC^*(a)$
28.  $CER \leftarrow CER(a) + CE^*(a)$
29.  $j \leftarrow j + 1$

#### 4.7.3. CorrelaciónDeVías(a)

Calcula en el atributo extra  $COR(a)[b][t]$  el índice de correlación de una flecha dada con todas las otras flechas de la gráfica en el tiempo  $t$ . Calcula en  $COR^*(a)[b]$  el índice de correlación entre  $a$  y cualquier flecha  $b$  durante la jornada completa. Requiere de una memoria extra de efectividad  $ME(a)[\tau][t]$ . El tiempo de ejecución es de orden  $\mathcal{O}(\|F(G)\| \cdot \Lambda)$

1.  $\triangleright$  Variables locales:
2. variable  $i, t, k \in \mathbb{N}$
3. variable  $b \in \mathbf{FLECHA}$
4. variable  $\tau, T[] \in \mathbf{USUARIO}$
5.  $\text{UnionConj}(\mathfrak{I}_a(a)[*], \mathfrak{I}_a(b)[*], T[*])$
6. **for**  $i \leftarrow 1$  **to**  $\text{CARDINAL}(F(G)[*])$
7.  $b \leftarrow F(G)[*]$
8. **for**  $t \leftarrow 1$  **to**  $\Lambda$
9.  $COR(a)[ind(b)][t] \leftarrow 0$

- 
10.       **for**  $k \leftarrow 1$  **to**  $\text{CARDINAL}(T[*])$
  11.              $\tau \leftarrow T[k]$
  12.             **if**  $ME(a)[ind(\tau)][t] < 0$
  13.               **else**
  14.                      $COR(a)[ind(b)][t] \leftarrow ME(a)[ind(\tau)][t] \times COR(a)[ind(b)][t] + \text{máx}\{0, ME(b)[ind(\tau)][t]\}$
  15.              $k \leftarrow k + 1$
  16.        $COR(a)[ind(b)][t] \leftarrow \frac{COR(a)[ind(b)][t]}{\text{CARDINAL}(T[*])}$
  17.        $COR^*(a)[ind(b)] \leftarrow COR^*(a)[b] + COR(a)[ind(b)][t]$
  18.        $t \leftarrow t + 1$
  19.        $COR^*(a)[ind(b)] \leftarrow \frac{COR^*(a)[ind(b)]}{\Lambda}$

# Apéndice A

## Dijkstra

En el algoritmo del capítulo 4, la componente que requiere mayor tiempo de ejecución, y que determina el orden del tiempo de ejecución, es el cálculo de las distancias  $\hat{d}_y^r$ . La naturaleza humana resuelve problemas de inmensa complejidad como dotarnos de orientación. El algoritmo que presentamos a continuación pretende simular el sentido de orientación de los conductores en la red vial. En el caso clásico se trata de una orientación puramente espacial, con la red como espacio fijo y que se concretiza en las distancias  $d_x^r(v)$  y  $d_y^r(v)$ . El caso dinámico requiere una orientación en una red que muta en respuesta al flujo que la recorre, misma que simulamos con la distancia dinámica  $d_y^r(v, t)$ . Finalmente la distancia relativa  $\hat{d}_y^r(v, t)$  provee al conductor de una orientación que responde a su percepción del tiempo en las vías de acuerdo a las características de éstas y el flujo que las recorre.

Dentro de este apéndice e independientemente de la notación general del trabajo definimos lo siguiente:

- Por *camino* entendemos una sucesión de flechas dirigidas  $C = (a_1, a_2, \dots, a_n)$  tales que  $a_i = (u, v) \Rightarrow a_{i+1} = (v, w)$  para  $1 \leq i < n$  donde  $u, v, w \in V(G)$ .
- Un camino  $C_{xv}$  de  $x$  hacia  $v$  es un camino que comienza en  $x$  y termina en  $v$ , de modo que  $a_1 = (x, u_1)$  y  $a_n = (u_{n-1}, v)$ .
- Denotamos  $C \subset C'$  cuando el camino  $C$  es parte del camino  $C'$ , de la misma forma  $C_{su} \subset C_{rv}$  implica que el camino  $C_{su}$  es la parte del camino  $C_{rv}$  que va de  $s$  hasta  $u$ , de forma que el camino  $C_{rv}$  comienza en  $r$  y durante su recorrido pasa por  $s$  (donde comienza el  $C_{su}$ ), luego por  $u$  (donde  $C_{su}$  termina) para llegar finalmente a  $v$ . Denotamos por  $V(C)$  y  $F(C)$  al conjunto de vértices y flechas, respectivamente, que forman parte del camino  $C$ .
- La *exvecindad* de un vértice  $v$  la denotamos por  $\Gamma^+(v) : V(G) \rightarrow 2^{V(G)}$ . Consiste de los

vértices que están unidos al vértice  $v$  por una flecha que sale de él, es decir,  $\Gamma^+(v) = \{w \in V(G) \mid \exists a \in F(G) \text{ que cumple } a = (v, w)\}$ , entonces,  $\Gamma^+(v) \subseteq V(G)$ .

## A.1. Problema clásico

Dada una digráfica  $G$  y una función, no negativa, de costo  $c: F(G) \rightarrow \mathbb{R}^+$ , el **problema** consiste en determinar, para cada  $v \in V(G)$ , el camino de menor costo (y el costo) desde un vértice distinguido  $x$ , llamado raíz, hacia  $v$ . Entendiendo el costo de un camino como la suma de los costos de las flechas que lo componen.

Un camino  $C_{xv}$  de  $G$  es un camino mínimo de  $v$  respecto a  $x$  si va de  $x$  hacia  $v$  y su costo es mínimo. El problema es entonces equivalente al de encontrar los caminos mínimos de la raíz  $x$  hacia todo  $v \in V(G)$ , por lo que también se le conoce como problema de los caminos mínimos. El algoritmo de Dijkstra, descrito por primera vez por Edsger Dijkstra en 1959 [W], resuelve el problema construyendo los caminos mínimos desde la raíz.

En el contexto de esta tesis, el costo de las flechas está dado por el tiempo de recorrido de las vías que representan; de modo que, el costo del camino  $C$  descrito anteriormente es  $tr(C) = \sum_{i=1}^n tr(a_i)$  y representa el tiempo de recorrido total del camino  $C$ . En este sentido, un camino mínimo  $C_{xv}$  es aquel que cumple:

$$\forall C_{xv}^* \text{ camino de } G, tr(C_{xv}) \leq tr(C_{xv}^*)$$

es decir, el camino más corto de  $x$  hacia  $v$ , en términos de tiempo de recorrido.

Definimos la *distancia desde la raíz*  $d_x: V(G) \rightarrow \mathbb{N}$  como el costo del camino mínimo de  $x$  hacia  $v$ , si existe un  $C_{xv}$  camino; se define  $d_x(v) = \infty$  si no hay ningún camino de  $x$  hacia  $v$ . Entonces, el algoritmo encuentra la distancia  $d_x(v) \forall v \in V(G)$  junto con los caminos mínimos que la determinan.

### A.1.1. Dijkstra clásico

El algoritmo asocia a cada vértice  $v$  un valor de costo  $d(v): V(G) \rightarrow \mathbb{R}^+$ , una etiqueta  $e(v): V(G) \rightarrow \{0, 1\}$  y un conjunto  $A(v): V(G) \rightarrow 2^{V(G)}$ , es decir  $A(v) \subseteq V(G)$ . El costo  $d(v)$  es el costo del camino más corto, de la raíz  $x$  hacia  $v$ , encontrado hasta ese momento. La etiqueta  $e(v)$  es 1 si el costo es permanente, es decir, si es el costo de un camino mínimo; y es 0 (provisional) si no se tiene esa certeza. El conjunto  $A(v)$  es el de los antecesores de  $v$ , es decir los vértices que están justo antes de  $v$  en los caminos que se han checado y tienen, hasta el momento, el menor costo. Al terminar el algoritmo  $d(v)$  indicará la distancia de  $v$  desde la raíz  $x$ , es decir, al final del algoritmo se cumple que  $d(v) = d_x(v)$ .

Al principio las distancias a la raíz no se conocen de modo que se establece  $d(v) = \infty \forall v \neq x$  y  $d(x) = 0$ . Cuando  $e(v) = 1$  significa que el algoritmo ha encontrado la distancia mínima de la raíz hacia el vértice  $v$ , en el otro caso,  $e(v) = 0$ , la distancia que el algoritmo asigna al vértice  $v$  es provisional, es decir, puede cambiar conforme éste avanza. Se establece en un principio  $e(v) = 0 \forall v \neq x$  y  $e(x) = 1$ . Definimos el conjunto  $Prov \subseteq V(G)$  como el conjunto de vértices con etiqueta provisional, entonces, en un principio se decreta  $Prov = V(G) \setminus \{x\}$ ; en este contexto, llamamos permanentes al conjunto  $Perm$  de vértices con etiqueta permanente (el complemento de  $Prov$ ).

El ciclo algorítmico consiste en encontrar el vértice más cercano a cualquiera de los vértices permanentes, determinar su distancia a partir de la de éstos y establecerlo como permanente. Para determinar el vértice más cercano a los permanente utiliza en cada ciclo algorítmico la flecha de menor costo, que tenga como vértice inicial un vértice permanente y como vértice final uno provisional. Los vértices permanentes son entonces el conjunto de vértices conocidos por  $x$  (permanentes), a los que se sabe como llegar desde  $x$  de la mejor manera. El algoritmo va haciendo más grande este conjunto incorporando los vértices inmediatos a los vértices permanentes, barriendo la gráfica hasta alcanzar todo vértice (para el que hay camino desde  $x$ ). Entonces el fin del algoritmo ocurre cuando todos los vértices son permanentes o cuando el siguiente vértice más cercano tiene un costo infinito.

Lo anterior se resume en:

### 1. Inicialización

- a)  $d(v) = \infty \forall v \neq x$
- b)  $d(x) = 0$
- c)  $e(v) = 0 \forall v \neq x$
- d)  $e(x) = 1$
- e)  $A(v) = \emptyset \forall v \in V(G)$
- f)  $Prov = V(G) \setminus \{x\}$
- g)  $z = x$

### 2. Paso algorítmico

- a) Para todo  $v \in \Gamma^+(z)$  establecer
 
$$d(v) = \begin{cases} d(v) & \text{si } d(v) < d(z) + tr(a) \\ d(z) + tr(a) \text{ y } A(v) = \{z\} & \text{si } d(v) > d(z) + tr(a) \\ d(v) \text{ y } A(v) = A(v) \cup \{z\} & \text{si } d(v) = d(z) + tr(a) \end{cases}$$
 donde  $a = (z, v) \in F(G)$
- b) Encontrar  $z^*$  tal que  $d(z^*) = \min\{d(v) \mid e(v) = 0, v \in V(G)\}$
- c) Si  $d(z^*) < \infty$ , establecer  $e(z^*) = 1, Prov = Prov \setminus \{z^*\}$

d) Establecer  $z = z^*$

### 3. Término

- a) Si  $Prov = \emptyset$       terminar
- b) Si  $d(z) = \infty$       terminar
- c) En otro caso      Repetir el ciclo algorítmico

La justificación del algoritmo se ve como caso particular del algoritmo dinámico de Dijkstra que se presenta más adelante en la sección A.2.2.

#### A.1.2. Distancias a flujo libre, $d_x^\tau, d_y^\tau$

Las distancias a flujo libre  $d_x^\tau(v)$  y  $d_y^\tau(v)$  se definen como el tiempo de recorrido de un camino mínimo  $C_{xv}$  que una a  $x$  con  $v$  o  $C_{vy}$  que una a  $v$  con  $y$ , respectivamente; donde la función de tiempo de recorrido corresponde al que resulta cuando el vehículo circula a la velocidad de flujo libre de las flechas, cuando la red está vacía:

$$d_x^\tau(v) = \min_{\{C_{xv} \text{ caminos}\}} \left\{ \sum_{i=1}^n \frac{\lambda(a_i)}{v(t_i)} \right\}$$

y

$$d_y^\tau(v) = \min_{\{C_{vy} \text{ caminos}\}} \left\{ \sum_{i=1}^m \frac{\lambda(b_i)}{v(t'_i)} \right\}$$

donde

$$\begin{aligned} C_{xv} &= \{a_0, a_1, \dots, a_n\} & a_0 &= (x, u_0), & a_n &= (u_{n-1}, v) \\ C_{vy} &= \{b_0, b_1, \dots, b_m\} & b_0 &= (v, u'_0), & b_m &= (u'_{m-1}, y) \end{aligned}$$

Dado un tipo  $\tau \in \mathbf{J}$  de usuario, la distancia  $d_x^\tau$  se obtiene directamente del algoritmo clásico de Dijkstra utilizando la gráfica  $G$  con el vértice origen de  $\tau$  como vértice raíz y la función de costos  $c(a) = \frac{\lambda(a)}{v(a)}$ . Para encontrar la distancia  $d_y^\tau$  necesitamos aplicar el algoritmo “al revés”, empezando con el vértice destino  $y$  del usuario  $\tau$  y encontrando “en reversa” los caminos mínimos de los vértices  $v$  hacia él, para lo cual es necesario voltear el sentido de todas las flechas de la gráfica.

A la flecha  $a$  con la dirección volteada la denotamos  $\tilde{a}$ , de forma que  $\tilde{a} = (v, u)$  si  $a = (u, v)$ . Entonces, definimos  $\tilde{G}$ , la *gráfica volteada* de  $G$ , como la gráfica que se obtiene de  $G$  volteando todas sus flechas.

$$F(\tilde{G}) = \tilde{F}(G) = \{\tilde{a} = (v, u) \mid \exists a = (u, v) \in F(G)\} \quad V(\tilde{G}) = V(G)$$

Entonces, para encontrar  $d_y^\tau$  basta correr el algoritmo de Dijkstra sobre la gráfica  $\tilde{G}$  con el vértice destino de  $\tau$  como raíz y la función de costos  $\tilde{c}(\tilde{a}) = c(a)$ .

## A.2. Problema dinámico

El tiempo de recorrido de una vía no es constante a lo largo del día o la noche, varía en función del flujo de la red. El algoritmo clásico de Dijkstra no es capaz de considerar estos cambios pues requieren una función que varíe conforme el tiempo pasa, sensible a la variación del flujo en la red.

Una ruta  $R$  es un camino en el que se considera el tiempo, es decir, una sucesión  $R = \{(a_0, t_0), (a_1, t_1), \dots, (a_n, t_n)\}$  de parejas  $(a, t)$  donde  $a \in F(G)$  y  $t \in \{t_0, \dots, N\}$  de forma que  $C_R = \{a_0, a_1, \dots, a_n\}$  es un camino y  $t_i \leq t_j$  si  $0 \leq i \leq j \leq n$ . Llamamos a  $C_R$  camino subyacente de  $R$ . Una  $R_{uv}$  ruta es aquella que comienza en el vértice  $u$  y termina en el  $v$ , sin importar en que tiempo inicia o termina; utilizamos la notación  $R \subset R'$  cuando la ruta  $R$  es parte (subruta) de la  $R'$ ; y  $V(R), F(R)$  para denotar los vértices y flechas (sin tiempos) que forman la ruta  $R$ . El costo o tiempo de recorrido de una ruta se define como la suma de los tiempos de recorrido de las flechas que la componen, en el tiempo en que se toman. Es decir,  $tr(R) = \sum_{i=1}^n tr(a_i, t_i)$ . Análogamente a los caminos, cuando una ruta  $R$  cumple que su tiempo de recorrido es mínimo, decimos que  $R$  es mínima.

Consideremos una ruta  $R = \{(a_0, t_0), \dots, (a_i, t_i), (a_{i+1}, t_{i+1}), \dots, (a_n, t_n)\}$  y una pareja intermedia  $(a_i, t_i)$ . Llamemos  $u$  al vértice inicial de  $a_i$  y  $v$  al final, de modo que  $a = (u, v)$ . Como  $R$  es ruta,  $v$  es el vértice inicial de  $a_{i+1}$  y se llega a él en el tiempo  $t_{i+1}$ . Por lo anterior sabemos que, en la ruta  $R$ , el tiempo de recorrido de  $u$  hacia  $v$  que corresponde a la flecha  $a_i$ , es  $t_{i+1} - t_i$ . Cuando este tiempo marcado por la ruta coincide con el de la función  $tr$  de tiempo de recorrido, decimos que  $R$  es una  $tr$ -ruta. De otra forma, una  $tr$ -ruta  $R = \{(a_0, t_0), (a_1, t_1), \dots, (a_n, t_n)\}$  es aquella que cumple que  $t_i + tr(a_i, t_i) = t_{i+1}$ . Viendo a las  $tr$ -rutas como conjuntos utilizamos el operador  $\cup$  para concatenar rutas de forma que:

$$\begin{aligned} R &= \{(a_0, t_0), \dots, (a_n, t_n)\}, a_n = (u, v), \\ S &= \{(b_0, s_0), \dots, (b_m, s_m)\}, b_0 = (v, w) \text{ y} \\ s_0 &= t_n + tr(a_n, t_n) \\ &\Rightarrow \\ R \cup S &= \{(a_0, t_0), \dots, (a_n, t_n), (b_0, s_0), \dots, (b_m, s_m)\} \end{aligned}$$

Dicho lo anterior el problema es: dadas una digráfica  $G$ , un periodo discreto de tiempos  $\{t_0, \dots, N\} \subset \mathbb{N}$  y una función positiva de tiempo de recorrido  $tr: F(G) \times \{t_0, \dots, N\} \rightarrow \mathbb{N}$ , determinar para cada  $v \in V(G)$  la  $tr$ -ruta de recorrido mínimo (y el tiempo de recorrido) desde un vértice distinguido  $x$ , llamado raíz, hacia todos los otros vértices  $v \in V(G)$ , comenzando en el tiempo  $t_0$ .

### A.2.1. Condición de fila

El *principio de buena construcción*, fundamental para el algoritmo de Dijkstra, dice:

Para toda  $x \in V(G)$  y todo  $v \in V(G)$  alcanzable desde  $x$ , *existe* un camino mínimo  $C_{xv}$  de  $x$  a  $v$  tal que, si  $C_{xu} \subset C_{xv}$  es un subcamino de  $x$  hacia  $u$  entonces  $C_{xu}$  es mínimo.

La importancia del principio es asegurar que, habiendo encontrado un camino mínimo  $C_{xu}$  para  $u$ , se puede construir un camino mínimo  $C_{xv}$  para  $v$  de forma inductiva, añadiendo una flecha al extremo del camino  $C_{x,u}$ . Enunciamos formalmente el principio, para la gráfica  $G$  y la función  $tr$ , como sigue<sup>1</sup>:

$$\begin{aligned} \forall v \in V(G) : & \\ & \exists C_{xv} \text{ camino de } G \\ \Rightarrow & \\ & \exists C_{xv} \text{ tal que } \forall u \in V(C_{xv}), \text{ el camino } C_{xu} \subset C_{xv} \text{ es mínimo en } G \end{aligned}$$

Decimos que un camino  $C_{xv}$  está *bien construido*<sup>2</sup> si  $\forall u \in V(C_{xv})$  tal que  $C_{xu} \subset C_{xv}$  se tiene que  $C_{xu}$  es camino mínimo de  $G$ . De la misma forma, decimos que,  $v \in V(G)$  cumple el principio de buena construcción si existe una ruta bien construida de  $x$  hacia  $v$ . De este modo el principio se cumple si todo vértice de  $G$  lo cumple. El principio de buena construcción y las definiciones relacionadas a él aplican en el caso dinámico<sup>3</sup> considerando las  $tr$ -rutas en lugar de los caminos.

En el caso clásico basta que la función de costos sea **no**-negativa para que se cumpla el principio. En el caso dinámico es posible esperar en los vértices un cierto tiempo y reanudar el recorrido (o tomar una ruta más larga para llegar más tarde a cierto vértice) de forma provechosa: cuando el tiempo de recorrido disminuye en más del tiempo que se espera a que lo haga (lo que no sucede en el caso clásico pues no varía el costo). Por ejemplo, digamos que en el tiempo 0 el recorrido para cruzar la flecha  $a$  dura 5 y en el tiempo 2 dura 1; en ese caso una ruta  $R_1$  que espera (o llega tarde) 2 tiempos para tomar la flecha  $a$  tendrá una duración de recorrido de  $2 + 1 = 3$  mientras que una ruta  $R_2$  que no espera durará 5. Ésto significa que no es posible construir la ruta  $R_1$  a partir de la  $R_2$ , suponiendo que ambas rutas son mínimas y únicas<sup>4</sup>, lo que contradice el principio de buena construcción.

<sup>1</sup>Se especifica la función  $tr$  pues la minimalidad a la que se refiere la definición es respecto a esta función, como función de costo sobre las flechas.

<sup>2</sup>El hecho de que un camino esté *bien construido* implica que es mínimo, pues  $C_{xv} \subset C_{xv}$ .

<sup>3</sup>Ver definición formal en la parte (b) de A.3

<sup>4</sup>Este argumento no es preciso, las hipótesis requieren consideraciones técnicas más detalladas que van más allá del ejemplo simple al que el argumento se refiere. El contraejemplo preciso del ejemplo se obtiene de la gráfica formada únicamente por la flecha  $a$  y sus vértices extremos, utilizando la función  $tr = \{(0, 5), (1, 4), (2, 1), (3, 1), (4, 1), (5, 1)\}$  como función de costos.

Para que falle el *principio de buena construcción* es necesario y suficiente que exista una flecha  $a = (u, v)$  para la que sea más provechoso esperar antes de tomarla, como en el ejemplo anterior. Esto contradice la práctica; si dos vehículos que viajan a la misma velocidad y utilizan el mismo carril entran a una vía con cierto tiempo de diferencia, aquel que entró antes saldrá primero<sup>5</sup>. Introducimos esta característica como condición sobre la función de tiempo de recorrido. La llamamos *condición de fila* pues es una característica propia de éstas: “el que llega primero sale primero”. En términos de la función  $tr$  esta característica se traduce como:

$$t + tr(a, t) \leq t + k + tr(a, t + k), \forall k \geq 0 \quad (\text{A.1})$$

o, lo que es lo mismo,

$$tr(a, t) - k \leq tr(a, t + k), \forall k \geq 0 \quad (\text{A.2})$$

Por ejemplo, si un vehículo  $V_1$  toma la flecha  $a$  en el tiempo  $t$ , saldrá de ella en el tiempo  $t + tr(a, t)$ , que corresponde al lado izquierdo de la desigualdad. Análogamente, si un segundo vehículo  $V_2$  toma la flecha  $a$  en el tiempo  $t + k$ ,  $k$  pasos después que  $V_1$ , saldrá de ella en el tiempo  $t + k + tr(a, t + k)$ , que corresponde al lado derecho de la desigualdad; la condición establece que: si  $V_2$  entró en  $a$  después o al mismo tiempo que  $V_1$  (si  $k \geq 0$ ), entonces saldrá después o al mismo tiempo que  $V_1$ .

Dada nuestra definición de ruta (y camino), para incorporar la posibilidad de espera, definimos la gráfica  $G^\circ$  como la extensión de  $G$  que tiene en cada vértice una flecha dirigida hacia sí mismo, un lazo:

$$V(G^\circ) = V(G) \quad y \quad F(G^\circ) = F(G) \cup \{(u, u) \mid u \in V(G)\}$$

Extendemos, abusando de la notación, la función  $tr$  a los lazos  $l$ , definiendo  $tr(l, t) = 1 \forall t \in \Lambda$ , donde  $l = (u, u)$ ,  $u \in V(G)$ , de forma que el tiempo de recorrido de todo lazo sea 1 para cualquier tiempo  $t$ . De esta manera nuestra definición de ruta (camino) permanece sin cambios e incorpora la espera (repetición) en un vértice mediante el uso de lazos. Definimos  $\overrightarrow{C}$  el *camino directo* de  $C$ , como el camino formado por la sucesión de flechas de  $C$  que no son lazos; así definimos el *camino directo subyacente* de la ruta  $R$ , como  $\overrightarrow{C}_R$ .

Veamos que, el hecho de que (a) la condición de fila se cumpla para todas las flechas de la gráfica en todo tiempo  $t$ ; implica que (b) se cumple el *principio de buena construcción* para la gráfica  $G^\circ$  y la función  $tr$ . Es decir:

---

<sup>5</sup>Las condiciones extras de velocidad y carril se siguen de que el tiempo de recorrido considera los vehículos indistintos.

$$\begin{array}{l}
(a) \left| \begin{array}{l} \forall a \in F(G^\odot) \text{ y } \forall t \in \Lambda, \\ tr(a, t) - k \leq tr(a, t + k), \forall k \geq 0 \end{array} \right. \\
\Rightarrow \\
(b) \left| \begin{array}{l} \forall t_0 \in \Lambda \forall x, v \in V(G^\odot), \\ \Rightarrow \exists C_{xv} \text{ camino de } G^\odot \\ \Rightarrow \exists R_{xv} \text{ que comienza en } t_0 \text{ y tal que } \forall u \in V(R_{xv}), \\ \text{la ruta } R_{xu} \subset R_{xv} \text{ es m\u00ednima en } G^\odot \end{array} \right. \quad (A.3)
\end{array}$$

En el inciso (b), el hecho de que exista  $C_{xv}$  camino de  $G^\odot$  implica la existencia de  $tr$ -rutas de  $x$  hacia  $v$ . Ya que la funci\u00f3n de costo de una ruta es positiva y el n\u00famero de  $tr$ -rutas finito, sabemos que existe una  $tr$ -ruta m\u00ednima. Entonces, para probar el inciso (b) basta checar que para todo  $v \in V(G^\odot)$ , dada la existencia de una  $tr$ -ruta m\u00ednima existe una bien construida, para todo v\u00e9rtice y tiempo iniciales.

Abusando de la notaci\u00f3n, definimos  $d_x(v, t_0)$ , la distancia din\u00e1mica de  $v$  relativa a  $x$ , como el tiempo de recorrido m\u00ednimo de las rutas que, empezando en el tiempo  $t_0$ , van de  $x$  hacia  $v$  (si no existe tal ruta,  $d_x(v, t_0) = \infty$ ). Probaremos la implicaci\u00f3n anterior por inducci\u00f3n sobre  $d_x(v, t_0)$ .

PRUEBA.

Base de Inducci\u00f3n,  $d_x(v, t_0) = 0$ . Corresponde a la ruta vac\u00eda que va de  $x$  a  $x$ . Por vacuidad se cumple el principio.

Hip\u00f3tesis de inducci\u00f3n,  $d_x(v, t_0) \leq n$ . Se cumple el principio de buena construcci\u00f3n para los v\u00e9rtices  $v$  que cumplan  $d_x(v, t_0) \leq n$

Paso inductivo,  $d_x(v, t_0) = n+1$ . Sea  $x, v \in V(G^\odot)$  tal que  $\exists R_{xv} = \{(a_0, t_0), (a_1, t_1), \dots, (a_m, t_m)\}$  de recorrido m\u00ednimo.

La \u00faltima flecha de la ruta  $R_{xv}$  no puede ser un lazo (que de serlo ser\u00eda  $a_m = (v, v)$ ), pues la ruta  $R'_{xv} = R_{xv} \setminus (a_m, t_m)$  cumplir\u00eda que  $tr(R'_{xv}) = tr(R_{xv}) - 1 < tr(R_{xv})$ , que contradice la minimalidad de  $R_{xv}$ . Por lo tanto, se  $u$  tal que  $a_m = (u, v)$  el v\u00e9rtice anterior a  $v$  en la ruta  $R_{xv}$  tal que  $u \neq v$ . Dicho v\u00e9rtice existe pues  $d_x(v, t_0) \geq 1$ .

Sea  $R_{xu}^* = \{(b_0, t_0), (b_1, t_1), \dots, (b_i, t_i)\}$  una ruta m\u00ednima hacia  $u$ , entonces:

$$tr(R_{xu}^*) \leq tr(R_{xu}) \leq tr(R_{xv}) - 1 \Rightarrow tr(R_{xu}^*) \leq n \Rightarrow d_x(u, t_0) \leq n \quad (A.4)$$

Por lo tanto y la hipótesis de inducción,  $u$  cumple el principio de buena construcción. Supongamos sin pérdida de generalidad que la misma  $R_{xu}^*$  es una ruta *bien construida* de  $u$ .

Consideremos la ruta  $S_{xv} = R_{xu}^* \cup (a_m, tr(R_{xu}^*))$  y  $k = tr(R_{xu}) - tr(R_{xu}^*)$ , donde  $R_{xu} \subset R_{xv}$ . Como  $R_{xu}^*$  es mínima, tenemos que  $tr(R_{xu}) \geq tr(R_{xu}^*)$  y por lo tanto  $k \geq 0$ . Por otro lado,

$$\begin{aligned} tr(R_{xv}) &= tr(R_{xu}) + tr(a_m, tr(R_{xu})) \\ &= tr(R_{xu}^*) + k + tr(a_m, tr(R_{xu}^*) + k) \\ &\geq tr(R_{xu}) + k + tr(a_m, tr(R_{xu}^*)) - k \quad \text{Cond. Fila} \\ &= tr(R_{xu}^*) + tr(a_m, tr(R_{xu}^*)) \\ &= tr(S_{xv}) \end{aligned}$$

$$\therefore tr(S_{xv}) \leq tr(R_{xv})$$

Pero  $R_{xv}$  es mínima, entonces,  $tr(S_{xv}) = tr(R_{xv})$ . Por lo tanto  $S_{xv}$  es mínima. Además, puesto que  $R_{xu}^*$  está bien construida, todas sus subrutas y ella misma son mínimas, de forma que  $S_{xv} = R_{xu}^* \cup (a_m, tr(R_{xu}^*))$ , está bien construida. Lo que concluye la prueba puesto que  $v, x$  y  $t_0$  son arbitrarios. □

### A.2.2. Dijkstra dinámico

El algoritmo de la versión dinámica de Dijkstra es, excepto por detalles obligados, igual al clásico. El detalle obligado aparece al momento de actualizar las distancias. En la versión clásica se establece, para todo  $v \in V(G)$ , el valor  $d(v) = \min\{d(v), d(z) + tr(a)\}$  donde  $a = (z, v) \in F(G)$ . Incorporando el tiempo como parámetro del costo, en la versión dinámica se establece  $d(v) = \min\{d(v), d(z) + tr(a, d(z))\}$ , es decir, el tiempo de recorrido en el momento  $d(z)$  en que el vehículo llega al vértice  $z$ , el inicial de la flecha  $a$ .

Dicho lo anterior el algoritmo dinámico de Dijkstra es:

#### 1. Inicialización

- a)  $d(v) = \infty \forall v \neq x$
- b)  $d(x) = t_0$
- c)  $e(v) = 0 \forall v \neq x$
- d)  $e(x) = 1$
- e)  $A(v) = \emptyset \forall v \in V(G)$

$$f) Prov = V(G) \setminus \{x\}$$

$$g) z = x$$

## 2. Paso algorítmico

a) Para todo  $v \in \Gamma^+(z)$  establecer

$$d(v) = \begin{cases} d(v) & \text{si } d(v) < d(z) + tr(a, d(z)) \\ d(z) + tr(a, d(z)) \text{ y } A(v) = \{z\} & \text{si } d(v) > d(z) + tr(a, d(z)) \\ d(v) \text{ y } A(v) = A(v) \cup \{z\} & \text{si } d(v) = d(z) + tr(a, d(z)) \end{cases}$$

donde  $a = (z, v) \in F(G)$

b) Encontrar  $z^*$  tal que  $d(z^*) = \min\{d(v) \mid e(v) = 0, v \in V(G)\}$

c) Si  $d(z^*) < \infty$ , establecer  $e(z^*) = 1$ ,  $Prov = Prov \setminus \{z^*\}$

d) Establecer  $z = z^*$

## 3. Término

a) Si  $Prov = \emptyset$       terminar

b) Si  $d(z) = \infty$       terminar

c) En otro caso      repetir el ciclo algorítmico

La versión clásica de Dijkstra es el caso particular de la dinámica en el que la función de costos es constante en el tiempo. Para probarlo basta ver que una función de costos constante en el tiempo respeta la condición de la fila.

Ser constante en el tiempo significa que  $tr(a, t) = tr(a, t + k) \forall t, t + k \in \Lambda$ , entonces  $\forall k \geq 0$  se cumple:

$$\begin{aligned} tr(a, t) &\leq tr(a, t) + k \\ &= tr(a, t + k) + k \end{aligned} \quad \Rightarrow \quad tr(a, t) - k \leq tr(a, t + k)$$

Que es exactamente la condición de fila.

Veamos que efectivamente este algoritmo encuentra el tiempo mínimo de recorrido desde el origen  $x$  hacia cualquier otro vértice  $v$ . Para la prueba necesitamos los siguientes lemas:

**Lema A.1.** *Dadas una gráfica  $G$  y  $R, S$  rutas de  $G$ , se cumple*

$$R \subset S \Rightarrow tr(R) \leq tr(S)$$

PRUEBA. Sean

$$S = ((a_1, t_1), \dots, (a_i, t_i), \dots, (a_{i+k}, t_{i+k}), \dots, (a_n, t_n))$$

y

$$R = ((a_i, t_i), (a_{i+1}, t_{i+1}), \dots, (a_{i+k}, t_{i+k}))$$

con  $1 \leq i$  e  $i + k \leq n$ , de modo que  $R \subset S$ . Entonces:

$$tr(S) = tr(a_1, t_1) + \dots + tr(a_{i-1}, t_{i-1}) + tr(R) + tr(a_{i+k}, t_{i+k}) + \dots + tr(a_n, t_n)$$

y como  $tr(a, t) > 0 \forall (a, t) \in F(G) \times \Lambda$ , tenemos:

$$tr(S) \geq tr(R)$$

□

**Lema A.2.** *A lo largo del algoritmo:*  $\forall v \in V(G), e(v) = 1 \Rightarrow d(v) < \infty$

PRUEBA. Al inicio del algoritmo,  $\forall v \in V(G) \setminus \{x\}$ , se tiene  $e(v) = 0, d(v) = \infty$ , es decir, todos los vértices excepto  $x$  son provisionales y su valor de costo  $d(v)$  se encuentra establecido como infinito. El valor de la etiqueta se modifica (de  $e(v) = 0$  a  $e(v) = 1$ ) en el paso  $2c$  de cada ciclo, solamente para el vértice  $z^*$  y sólo cuando  $d(z^*) < \infty$ . Por otro lado,  $d(v)$  no se modifica si  $e(v) = 1$ , entonces para cualquier vértice  $v$  con etiqueta  $e(v) = 1$  podemos asegurar que  $d(v) < \infty$  desde que  $e(v) = 1$  y hasta el final del algoritmo. □

**Teorema A.1.** *Al concluir, el algoritmo dinámico de Dijkstra, los costos  $d(v)$  de los vértices coinciden con la distancia  $d_x(v, t_0)$ , de forma que:*

$$d(v) = d_x(v, t_0) \forall v \in V(G)$$

PRUEBA.

Supongamos que al terminar cada ciclo  $n$  se cumple que:

$$(a) \forall v \in V(G) \quad d(v) = \min \left( \left\{ tr(R_{xv}) \mid V(R_{xv}) \subset Perm \setminus \{z\} \cup \{v\} \right\} \cup \{\infty\} \right)$$

Es decir, el costo  $d(v)$  de todo vértice  $v$  es el costo mínimo de una ruta  $R_{xv}$  que va de  $x$  a  $y$  pasando únicamente por  $v$  y vértices permanentes (entre los que se excluye a  $z$  por ser el vértice permanente más reciente). En caso de que no haya rutas como tales el costo se define infinito.

$$(b) d(v) = d_x(v, t_0) \forall v \in V(G) \text{ tal que } e(v) = 1$$

En particular (a) se cumple al terminar el último ciclo de manera que, para los vértices permanentes obtenemos directamente la igualdad que buscamos. El algoritmo termina si todos los vértices son permanentes, en cuyo caso habríamos terminado, o cuando  $d(z) = \infty$ . Por construcción, al terminar el algoritmo  $d(z) = d(z^*) = \min\{d(v) \mid e(v) = 0, v \in V(G)\}$ , de modo que  $d(z) = \infty$  implica

que  $d(v) = \infty \forall v \in Prov$ . Por (b) sabemos entonces que no hay rutas que liguen a ninguno de los vértices provisionales con el raíz, de modo que, por definición  $d_x(v, t_0) = \infty = d(v)$ .

Dicho lo anterior basta probar (a) y (b), lo que haremos por inducción sobre el número de ciclos del algoritmo.

- **Base de inducción.** En el ciclo 0, la inicialización: el único vértice etiquetado como permanente, y el único de costo finito es  $x$ . El conjunto  $Perm \setminus \{z\} \cup \{x\} = \{x\} \setminus \{x\} \cup \{x\} = \{x\}$ , por lo que la única ruta posible es la vacía  $R_{xx} = \emptyset$  cuyo costo es nulo. Dado que  $R_{xx} = \emptyset$ , para cualquier otra ruta  $R_{xx}^*$  se tiene que  $R_{xx} \subset R_{xx}^*$ , y por el lema A.1 obtenemos  $0 = tr(R_{xx}) \leq tr(R_{xx}^*)$ , por lo tanto es mínima y se cumple  $d_x(x) = d(x)$ .

- **Hipótesis de inducción.** Supongamos validas (a) y (b) para toda  $k \leq n$

- **Paso inductivo.** Veamos que se cumplen las hipótesis para  $n + 1$ .

(a) Sea  $v \in V(G)$ , consideremos una ruta  $R_{xv}$  tal que  $V(R_{xv}) \subset Perm \setminus \{z^*\} \cup \{v\}$ . Supongamos que el algoritmo acaba de realizar el paso 2c, de manera que  $z$  es el nuevo vértice permanente del ciclo anterior y  $z^*$  el del ciclo  $n + 1$ .

- Si  $z \notin V(R_{xv})$ :

$$\begin{aligned} & V(R_{xv}) \subset (Perm \setminus \{z^*\}) \setminus \{z\} \cup \{v\} \\ H.I. \Rightarrow & \\ & d(v) \leq tr(R_{xv}) \end{aligned}$$

Obs: el conjunto  $Perm$  del final del ciclo  $n$ , donde aplica la H.I., no contenía a  $z^*$ , por lo que es igual al  $Perm \setminus \{z^*\}$  del ciclo  $n + 1$  después del paso 2c.

- Si  $z \in R_{xv}$ : sea  $u$  el vértice anterior a  $v$  en  $R_{xv}$ , es decir:

$$R_{xu} \cup (a, tr(R_{xu})) = R_{xv} \text{ con } a = (u, v)$$

Por el lema A.1

$$u \in Perm \Rightarrow d(u) < \infty$$

- Supongamos  $u \neq z$ , por hipótesis de inducción

$$\exists R_{xu}^* \text{ tal que } V(R_{xu}^*) \subset Perm \setminus z \text{ y } d(u) = tr(R_{xu}^*) \leq tr(R_{xu})$$

Sea  $R_{xv}^* = R_{xu}^* \cup (a, tr(R_{xu}^*))$  y  $k = tr(R_{xu}) - tr(R_{xu}^*)$ , entonces:

$$\begin{aligned} tr(R_{xv}^*) &= tr(R_{xu}^*) + tr(a, tr(R_{xu}^*)) \\ &= tr(R_{xu}) + k + tr(a, tr(R_{xu}) + k) \quad A.2 \Rightarrow \\ &\leq tr(R_{xu}) + k + tr(a, tr(R_{xu})) - k \\ &= tr(R_{xv}) \end{aligned}$$

El valor de  $d(v)$  sólo se modifica en el paso 2a del algoritmo, y en caso de ser modificado decrece. Dado que el vértice  $u$  fue el vértice  $z$  para algún ciclo  $n^* < n$  y como  $v \in \Gamma^+(u)$ , tenemos:

$$d(v) \leq d(u) + tr(a, d(u)) = tr(R_{xu}^*) + tr(a, tr(R_{xu}^*)) = tr(R_{xv}^*) \leq tr(R_{xv})$$

- o Supongamos que  $u = z$ . Como en el ciclo anterior  $d(v) = \min\{tr(S_{xv}) \mid V(S_{xv}) \subset Perm \setminus \{z, z^*\} \cup \{v\}\}$ , tenemos:

$$\begin{aligned} d(v) &= \min \left( \left\{ tr(S_{xv}) \mid V(S_{xv}) \subset Perm \setminus \{z, z^*\} \cup \{v\} \right\} \cup \right. \\ &\quad \left. \left\{ d(z) + tr(b, d(z)) \mid b = (u, v) \right\} \right) \\ &= \min\{tr(S_{xv}) \mid V(S_{xv}) \subset Perm \setminus \{z^*\} \cup \{v\}\} \end{aligned}$$

Lo anterior puesto que  $d(z)$  y las flechas  $b = (u, v)$  toman en cuenta las rutas  $S_{xv}$  con  $V(S_{xv}) \subset Perm \setminus \{z^*\} \cup \{v\}$  que pasan por  $z$ . Después del paso 2d nos queda:

$$d(v) = \min \left\{ tr(S_{xv}) \mid V(S_{xv}) \subset Perm \setminus \{z\} \cup \{v\} \right\}$$

Retomando los otros casos y dado que  $R_{xv}$  es arbitraria, tenemos que  $d(u) \leq tr(r_{xv})$  implica que el inciso (a) se cumple para  $n + 1$ .

- (b) Sea  $z^*$  como se define en el paso 2b del algoritmo, de modo que:

$$d(z^*) = \min\{d(v) \mid e(v) = 0, v \in V(G)\}$$

Por la hipótesis de inducción,

$$d(z^*) = \min \left\{ tr(R_{xv}) \mid v \in V(G), V(R_{xv}) \subset Perm \setminus \{z\} \cup \{v\}, e(v) = 0 \right\}$$

Supongamos que existe  $R_{xz^*}$ . Sea  $v$  el primer vértice provisional (ó  $z$ ) de  $R_{xz^*}$ , es decir:

$$e(v) = 0, R_{xv} \subset R_{xz^*} \text{ y } V(R_{xv}) \subset Perm \setminus z \cup \{v\}$$

Por el lema A.1

$$d(z^*) \leq tr(R_{xv}) \leq tr(R_{xz^*})$$

y como  $R_{xz^*}$  es arbitraria,

$$d(z^*) = \min\{tr(R_{xz^*}) \mid R_{xz^*} \text{ ruta de } G\} = d_x(z^*)$$

Como  $z^*$  es el único vértice para el que cambia la etiqueta, por H.I. tenemos:

$$\begin{aligned} d(v) &= d_x(v) \quad \forall v \neq z^* \\ \Rightarrow \\ d(v) &= d_x(v) \quad \forall v \in V(G) \end{aligned}$$

□

### A.2.3. Distancia dinámica $d_y^r(v, t)$

Definimos el *tiempo volteado*  $\tilde{t}$  como  $\tilde{t} = \Lambda - t$ . Dado un usuario  $\tau \in \mathbb{J}$  y un tiempo inicial  $\tilde{t}_{fin} \in \Lambda$  aplicamos Dijkstra dinámico sobre la *gráfica volteada*  $\tilde{G}$  con el vértice  $y$ , destino de  $\tau$ , como raíz y  $\tilde{tr}(a, t)$ , la función *volteada* de  $tr$ , como función de costo. Dicho algoritmo encuentra, para  $\tilde{t}_{fin}$  fijo, una *distancia parcial*<sup>6</sup>  $\mathbf{d}(v, \tilde{t}_{fin})$  que se define como el menor tiempo de recorrido de las  $tr$ -rutas que van de  $v$  hacia  $y$  llegando a éste último en el tiempo  $\tilde{t}_{fin}$ ; se define como infinito ( $\mathbf{d}(v, \tilde{t}_{fin}) = \infty$ ) si no existe una ruta como la descrita.

Recabados todos los valores posibles de  $\mathbf{d}(v, \tilde{t}_{fin})$  al correr Dijkstra dinámico para todo  $\tilde{t}_{fin} \in \Lambda$ . Construimos la distancia  $d_y^r(v, t)$  como:

$$d_y^r(v, t) = \text{mín}\{t_{fin} - t \mid t_{fin} - \mathbf{d}(v, \tilde{t}_{fin}) \geq t\} \quad (\text{A.5})$$

Esta sección demuestra que esta fórmula coincide con la definición de la distancia  $d_y^r$ .

La función *volteada*  $\tilde{tr}$  requiere mayores consideraciones que su homóloga en el caso clásico; por definición debe decirnos el tiempo que tardó un vehículo en cruzar la flecha  $\tilde{a}$  si terminó de hacerlo en el tiempo  $t = \Lambda - \tilde{t}$  (pues el algoritmo debe correrse “al revés”, del tiempo final  $\Lambda$  hacia el inicial 1); sin embargo no siempre existe. Digamos que un vehículo  $V_1$  llega a la flecha  $a = (u, v)$  en el tiempo 1 con  $tr(a, 1) = 2$ , y otro vehículo  $V_2$  llega en el tiempo 2 con  $tr(a, 2) = 1$  (lo cual cumple la condición de fila), entonces ambos vehículos llegarán al vértice  $v$  en el tiempo 3.  $V_2$  alcanzó a  $V_1$ . ¿Qué valor debería tener  $\tilde{tr}(\tilde{a}, \Lambda - 3)$ ? (¿qué valor debería tener  $\tilde{t}$  si  $t = 3$ ?). Por otro lado, si  $V_1$  llega en el tiempo 1 con  $tr(a, 1) = 2$  y  $V_2$  en el tiempo 2 con  $tr(a, 2) = 3$  obtenemos un hueco.  $V_1$  terminará en el tiempo 3 y  $V_2$  en el 5, nadie en el 4. De nuevo  $\tilde{t}$  para  $t = 4$  y  $\tilde{tr}(\tilde{a}, \Lambda - 4)$  nos quedan indefinidos.

Definimos entonces  $\tilde{tr}(\tilde{a}, \tilde{t})$  como:

$$\tilde{tr}(\tilde{a}, \tilde{t}) = \begin{cases} t - s \text{ con } s = \text{máx}\{q \in \Lambda \mid q + tr(a, q) \leq t\} & \text{si existe} \\ \infty & \text{en otro caso} \end{cases}$$

donde  $\tilde{t} = \Lambda - t$ . Siguiendo los ejemplos anteriores, si (en el primero) añadimos que  $tr(a, 3) = 1$ , tenemos que  $\tilde{tr}(\tilde{a}, \Lambda - 3) = 1$ , que implica elegir el tramo de ruta de  $V_2$ ; en el segundo ejemplo, considerando la condición de cola, tenemos  $\tilde{tr}(\tilde{a}, \Lambda - 4) = 3$  de modo que los dos casos problemáticos están bien definidos para  $\tilde{tr}$ . Ya que aplicamos el algoritmo dinámico de Dijkstra con  $\tilde{tr}$ , es necesario que ésta cumpla la *condición de fila*, veamos que lo hace:

$$\begin{aligned} \tilde{t} + k + \tilde{tr}(\tilde{a}, \tilde{t} + k) &= \tilde{t} + k + \tilde{tr}(\tilde{a}, \widetilde{(t - k)}) \\ &= (\Lambda - t) + k + ((t - k) - s) \\ &= \Lambda - s = \tilde{s} \end{aligned}$$

<sup>6</sup>La función de distancia  $\mathbf{d}(v, \tilde{t}_{fin})$  es auxiliar, tiene el parámetro de tiempo *volteado* de manera que  $\mathbf{d}(v, \tilde{0})$  es el tiempo mínimo de recorrido de  $v$  hacia  $y$  a modo de llegar a  $y$  en el tiempo  $\tilde{0} = \Lambda - 0$ . Sirve únicamente para calcular la distancia dinámica  $d_y^r(v, t)$ .

Donde  $s = \text{máx}\{q \in \Lambda \mid q + tr(a, q) \leq t - k\}$ . Por otro lado,

$$\tilde{t} + \tilde{tr}(\tilde{a}, \tilde{t}) = (\Lambda - t) + (t - r) = \Lambda - r = \tilde{r}$$

donde

$$r = \text{máx}\{q \in \Lambda \mid q + tr(a, q) \leq t\}.$$

Puesto que  $k > 0$ ,

$$s \in \{q \in \Lambda \mid q + tr(a, q) \leq t\}$$

Se sigue entonces que  $s \leq r$  y por lo tanto  $\tilde{r} \leq \tilde{s}$ , de donde

$$\tilde{t} + \tilde{tr}(\tilde{a}, \tilde{t}) = \tilde{r} \leq \tilde{s} = \tilde{t} + k + \tilde{tr}(\tilde{a}, \tilde{t} + k)$$

que es exactamente la condición de fila para  $\tilde{tr}$ . Finalmente, requerimos las siguientes definiciones y el lema posterior a ellas.

Decimos que las rutas  $R$  y  $S$  cumplen  $R \preceq S$  si sus caminos subyacentes son iguales, sin ciclos y  $R$  va detrás o igual que  $S$  en todo momento. Formalmente:

$$R \preceq S \leftrightarrow \begin{cases} C_R = C_S = \vec{C}_R = \vec{C}_S \\ \forall (u, t_i) \in R, (v, t_j) \in S \text{ se tiene } u = v \rightarrow t_i \leq t_j \end{cases}$$

Decimos que  $R \prec S$  si  $R$  va estrictamente detrás de  $S$ . Formalmente:

$$R \prec S \leftrightarrow \begin{cases} C_R = C_S = \vec{C}_R = \vec{C}_S \\ \forall (u, t_i) \in R, (v, t_j) \in S \text{ se tiene } u = v \rightarrow t_i < t_j \end{cases}$$

**Lema A.3.** Dado  $k > 0$  y  $\tilde{R}$ ,  $S$  y  $S^+$  rutas de  $\tilde{G}$ , tales que:

$$\begin{aligned} \tilde{R} &= \{(\tilde{a}_n, \tilde{t}_n), \dots, (\tilde{a}_0, \tilde{t}_0)\} && \text{donde } t_{i+1} = t_i + tr(a_i, t_i) \\ S &= \{(\tilde{a}_n, \tilde{t}'_n), \dots, (\tilde{a}_0, \tilde{t}'_0)\} && \text{donde } \tilde{t}'_n = \tilde{t}_n \text{ y } \tilde{t}'_i = \tilde{t}'_{i+1} + \tilde{tr}(\tilde{a}_{i+1}, \tilde{t}'_{i+1}) \\ S^+ &= \{(\tilde{a}_n, \tilde{t}''_n), \dots, (\tilde{a}_0, \tilde{t}''_0)\} && \text{donde } \tilde{t}''_n = \tilde{t}_n + k \text{ y } \tilde{t}''_i = \tilde{t}''_{i+1} + tr(\tilde{a}_{i+1}, \tilde{t}''_{i+1}) \end{aligned}$$

se cumple:

$$S \preceq \tilde{R} \prec S^+$$

PRUEBA.

- Por hipótesis,  $\tilde{t}'_n = \tilde{t}_n < \tilde{t}''_n$ . Supongamos que **no** se cumple el lema, de modo que:

$$\exists m \in \mathbb{N}, 0 \leq m \leq n, \text{ tal que } \tilde{t}'_m > \tilde{t}_m \text{ ó } \tilde{t}_m \geq \tilde{t}''_m$$

Consideramos, sin pérdida de generalidad, que  $m$  es el primer índice para el que ésto sucede<sup>7</sup>. Entonces,

$$\tilde{t}'_{m+1} \leq \tilde{t}_{m+1} < \tilde{t}''_{m+1}$$

<sup>7</sup>Considerando  $n$  como el primer índice y 0 como el último, en el sentido en que se desarrollan las rutas en cuestión.

- Como  $t_{m+1} = t_m + tr(a_m, t_m)$  tenemos que,

$$t_m \leq s = \text{máx}\{r \in \Lambda \mid r + tr(a_m, r) \leq t_{m+1}\}$$

y por la condición de fila<sup>8</sup>

$$t_m > s' = \text{máx}\{q \in \Lambda \mid q + tr(a_m, q) < t_{m+1}\}$$

$$\therefore s' < t_m \leq s$$

$$\therefore \tilde{s} \leq \tilde{t}_m < s'$$

- Por definición de  $S$  y  $\tilde{tr}$ ,

$$\begin{aligned} \tilde{t}'_m &= \tilde{t}'_{m+1} + \tilde{tr}(\tilde{a}_{m+1}, \tilde{t}'_{m+1}) \\ &= \Lambda - t'_{m+1} + t'_{m+1} - s \\ &= \Lambda - s \\ &= \tilde{s} \end{aligned}$$

- Por definición de  $S^+$  y  $\tilde{tr}$

$$\begin{aligned} \tilde{t}''_m &= \tilde{t}''_{m+1} + \tilde{tr}(\tilde{a}_{m+1}, \tilde{t}''_{m+1}) \\ &= \Lambda - t''_{m+1} + t''_{m+1} - s' \quad \text{Def. } \tilde{tr} \\ &= \Lambda - s' \\ &= \tilde{s}' \end{aligned}$$

- Por lo tanto

$$\tilde{t}'_m \leq \tilde{t}_m < \tilde{t}''_m$$

□

**Teorema A.2.** La distancia  $d_y^r(v, t)$ , definida como el tiempo de recorrido mínimo de una ruta de  $G$  que va de  $v$  hacia y partiendo en el tiempo  $t$ , cumple:

$$\forall (v, t) \in V(G) \times \Lambda \quad d_y^r(v, t) = \text{mín}\{t_{fin} - t \mid t_{fin} - \mathbf{d}(v, \tilde{t}_{fin}) \geq t\}$$

PRUEBA. Sea  $R_{vy} = \{(a_0, r_0), \dots, (a_n, r_n)\}$ ,  $tr$ -Ruta mínima de  $G$  y  $\tilde{R}_{yv} = \{(\tilde{a}_n, \tilde{r}_n), \dots, (\tilde{a}_0, \tilde{r}_0)\}$ , lo que llamamos la ruta *volteada*<sup>9</sup> de  $R_{vy}$ . Sea  $S = \{(\tilde{a}_n, \tilde{s}_n), \dots, (\tilde{a}_0, \tilde{s}_0)\}$ ,  $\tilde{tr}$ -Ruta con  $\tilde{s}_n = \tilde{r}_n$ .

<sup>8</sup>Si  $t_m \leq s' \Rightarrow \exists k \geq 0$  tal que  $t_m + k = s' \Rightarrow t_{m-1} = t_m + tr(a_m, t_m) \leq t_m + k + tr(a_m, t_m + k) = s' + tr(a, s') < t_{m-1}$

<sup>9</sup>Recordando  $\tilde{r} = \Lambda - r$  tenemos:  $\tilde{R} = \{(\tilde{a}_n, \Lambda - r_n), \dots, (\tilde{a}_0, \Lambda - r_0)\}$ .

$$\begin{aligned}
 \text{Lema A.3} & \Rightarrow S \preceq \tilde{R} \\
 \text{Definición } \preceq & \Rightarrow \tilde{s}_0 \leq \tilde{r}_0 \text{ y como } S \text{ comienza en } \tilde{s}_n = \tilde{r}_n \\
 \text{Construcción } S & \Rightarrow \tilde{tr}(S) = \tilde{s}_0 - \tilde{s}_n \leq \tilde{r}_0 - \tilde{r}_n = r_n - r_0 \\
 \text{Def. } \mathbf{d} \text{ y Teo.A.1} & \Rightarrow \mathbf{d}(v, \tilde{r}_n) \leq \tilde{tr}(S) \leq r_n - r_0 = tr(R) \\
 & \Rightarrow r_n - \mathbf{d}(v, \tilde{r}_n) \geq r_0 \text{ y como } tr(R) = r_n - r_0 \\
 tr(R) \text{ mínima} & \Rightarrow \min\{t - r_0 \mid r - \mathbf{d}(v, \tilde{r}) \geq r_0\} \leq r_n - r_0 = tr(R) = d_y^r(v, r_0)
 \end{aligned}$$

Supongamos que  $\min\{t - r_0 \mid t - \mathbf{d}(v, \tilde{t}) \geq r_0\} < d_y^r(v, r_0)$ , entonces:

$$\exists t \in \Lambda \text{ tal que } t - r_0 < d_y^r(v, r_0) \text{ y } t - \mathbf{d}(v, \tilde{t}) \geq r_0$$

Sea  $T_{yv} = \{(\tilde{b}_m, \tilde{t}_m), \dots, (\tilde{b}_0, \tilde{t}_0)\}$   $\tilde{tr}$ -ruta mínima de  $\tilde{G}$  tal que  $t_m = t$ , entonces

$$t_m - r_0 < d_y^r(v, r_0) = tr(R) = r_n - r_0 \text{ y } t_m - \mathbf{d}(v, \tilde{t}_m) \geq r_0$$

$$\begin{aligned}
 \Rightarrow t_m < r_n & \quad \text{y} \quad t_0 = t_m - (t_m - t_0) \\
 & = t_m - (\tilde{t}_0 - \tilde{t}_m) \\
 & = t_m - \tilde{tr}(T) \\
 & = t_m - \mathbf{d}(v, \tilde{t}_m) \geq r_0
 \end{aligned}$$

Sea  $R' = \{(b_0, r'_0), \dots, (b_m, r'_p)\}$  con  $r'_{i+1} = r'_i + tr(b_i, r'_i)$  y  $r'_0 = t_0$ . Sea  $\tilde{R}' = \{(\tilde{b}_p, \tilde{r}'_p), \dots, (\tilde{b}_0, \tilde{r}'_0)\}$  la ruta volteada de  $R'$ . Entonces  $\tilde{r}'_0 = \tilde{t}_0$ . Como  $R$  es mínima

$$r_n - r_0 = tr(R) \leq tr(R') = r'_p - r'_0 = r'_p - t_0 \leq r'_p - r_0$$

$$\Rightarrow r_n \leq r'_p$$

$$\Rightarrow t_m < r_n \leq r'_p$$

Sea  $k > 0$  tal que  $t_m + k = r'_p$ . Por el lema A.3 tenemos que  $\tilde{R}' \prec T$ , entonces

$$\tilde{r}'_0 < \tilde{t}_0 \Rightarrow t_0 < r_0$$

que es una contradicción.

□

### A.3. Problema relativo

Los distintos usuarios de una red no perciben igual el tiempo de recorrido de una vía. De acuerdo a las características de cada vía y a las preferencias de los usuarios sobre dichas características, el usuario preferirá pasar el tiempo de su viaje recorriendo una u otra. De otra forma, dado el tiempo de recorrido y las características de una vía, el usuario asignará a ésta un costo de acuerdo con sus preferencias. La función que lleva a cabo dicha asignación se denomina función de costo relativo.

La mejor ruta para cada usuario será aquella que tenga el mínimo costo relativo de acuerdo con sus preferencias. Nuestro nuevo problema es encontrar la ruta mínima y su costo de acuerdo a esta función. Formalmente el problema relativo es: dadas una digráfica  $G$ , un periodo discreto de tiempos  $\{t_0, \dots, N\} \subset \mathbb{N}$ , una función no negativa de tiempo de recorrido  $tr: F(G) \times \{t_0, \dots, N\} \rightarrow \mathbb{N}$  y una función no negativa de costo  $cr: F(G) \times \{t_0, \dots, N\} \rightarrow \mathbb{N}$ , determinar para cada  $v \in V(G)$  la  $tr$ -ruta de costo mínimo (y el costo) desde un vértice distinguido  $x$ , llamado raíz, hacia todos los otros vértices  $v \in V(G)$ , comenzando en el tiempo  $t_0$ .

Si la función de costo relativo es constante en el tiempo (y por lo tanto indiferente a los cambios de flujo de las flechas), el problema relativo se reduce al clásico. Al hablar del problema relativo nos referimos a una función dinámica de costo.

En el problema dinámico la función  $tr$  define el costo de las flechas en términos del tiempo que se invierte en recorrerlas, y al hacerlo determina la sucesión de tiempos de las  $tr$ -rutas. Es decir, determina para todas las rutas en que momento (tiempo) pasarán por cada vértice. En el problema presente el costo relativo de una  $tr$ -ruta (y las flechas que la componen) está dado por la función  $cr$  y las sucesión de tiempos de las rutas por la función  $tr$ . En el problema relativo el costo de cada  $tr$ -ruta depende de ambas funciones, la  $cr$  y la  $tr$ . Aún considerando el costo relativo como un tiempo de recorrido, el percibido por el conductor, los vehículos recorrerán cada flecha en el tiempo real ( $tr$ ) y no en el que perciben por sus preferencias.

Utilizando las funciones  $tr$  y  $cr$  podemos determinar el costo de todas las  $tr$ -rutas en  $G$ , sin embargo es posible que la ruta mínima no cumpla el principio de buena construcción (ver A.2.1) por lo que es invisible para el algoritmo dinámico. Dado ésto y para resolver el problema relativo, construimos a partir de la gráfica  $G$  lo que llamamos *gráfica escalonada* de  $G$ . Utilizando el algoritmo clásico de Dijkstra y esta nueva gráfica obtenemos los resultados que buscamos; sin embargo el tiempo de ejecución es mucho mayor.

### A.3.1. Dijkstra escalonado

Definimos la *gráfica escalonada*  $\mathfrak{E}(G, tr, \Lambda)$  de  $G$  respecto a una función de tiempo de recorrido  $tr: F(G) \times \{1, \dots, \Lambda\} \rightarrow \mathbb{N}$  de la siguiente manera: Los vértices de  $\mathfrak{E}(G, tr, \Lambda)$  consisten de parejas ordenadas  $(v, t)$  que representan, en términos de la red vial, un crucero en un momento preciso.

$$V(\mathfrak{E}(G, tr, \Lambda)) = V(G) \times \{1, \dots, \Lambda\}$$

En  $\mathfrak{E}(G, tr, \Lambda)$  hay una flecha  $a$  de  $(u, t)$  a  $(v, t^*)$  cuando en  $G$  existe una flecha  $f$  de  $u$  a  $v$  que cumple  $t + tr(f, t) = t^*$ .

$$F(\mathfrak{E}(G, tr, \Lambda)) = \{((u, t), (v, t^*)) \mid (u, v) \in F(G) \text{ y } t + tr((u, v), t) = t^*\}$$

Para cada tiempo  $t \in \{1, \dots, \Lambda\}$  tenemos una copia de  $G$  con todo y flechas, excepto porque el vértice final de ellas está en otro escalón (digamos, más arriba): Llamamos *escalones* a dichas copias de  $G$  pues asemejan la función de estos para las rutas de  $\mathfrak{E}(G, tr, \Lambda)$ ; en el mismo sentido nombramos a  $\mathfrak{E}(G, tr, \Lambda)$  la *gráfica escalonada* de  $G$ . Una característica importante de esta construcción es que los caminos en  $\mathfrak{E}(G, tr, \Lambda)$  corresponden exactamente a las  $tr$ -rutas en  $G$ , por lo que los llamamos  $tr$ -rutas y utilizamos indistintamente para  $G$  o  $\mathfrak{E}(G, tr, \Lambda)$ . Más aún la notación  $R = \{(a_0, t_0), (a_1, t_1), \dots, (a_n, t_n)\}$  de una  $tr$ -ruta en  $G$  corresponde exactamente a la del camino en  $\mathfrak{E}(G, tr, \Lambda)$  asociado a la  $tr$ -ruta  $R$ . En este contexto, definimos las flechas y vértices de una  $tr$ -ruta  $R$  como:

$$V(R) = \{(v, t) \in V(\mathfrak{E}(G, tr, \Lambda)) \mid (v, t) \text{ es final o inicial de } \bar{a} \in R \wedge \bar{a} \in F(\mathfrak{E}(G, tr, \Lambda))\}$$

Consideremos una función  $c$  de costo, sobre las flechas de  $G$ , que depende del tiempo:  $c: F(G) \times \{1, \dots, \Lambda\} \rightarrow \mathbb{N}$ . Por ejemplo, el consumo de combustible (que depende, entre otras cosas, de la congestión y ésta del tiempo). Esta función se extiende en  $\mathfrak{E}(G, tr, \Lambda)$  a una función  $\bar{c}: F(\mathfrak{E}(G, tr, \Lambda))$  definida como  $\bar{c}(\bar{a}_t) = c(a, t)$ , donde  $\bar{a}_t = ((u, t), (v, t^*))$ . Dado que  $\bar{c}$  es constante en el tiempo y toda ruta  $R$  en  $\mathfrak{E}(G, tr, \Lambda)$  es una  $tr$ -ruta, al correr Dijkstra clásico en  $\mathfrak{E}(G, tr, \Lambda)$  con  $\bar{c}$  como función de costo, obtenemos la  $tr$ -ruta<sup>10</sup> en la que se gasta menos combustible y el costo del cosumo.

El algoritmo en esta versión, que consiste en construir  $\mathfrak{E}(G, tr, \Lambda)$  a partir de  $G$  y correr sobre ella *Dijkstra Clásico* (con algunas modificaciones obligadas) se denomina *algoritmo escalonado de Dijkstra*. Este algoritmo encuentra, en la gráfica  $G$ , la  $tr$ -ruta de costo mínimo para una función  $c$  de costo definida sobre las flechas de  $G$  y dinámica en el tiempo.

<sup>10</sup>Por definición, el costo de una  $tr$ -ruta  $R$  en  $G$  es el mismo que el costo como camino en  $\mathfrak{E}(G, tr, \Lambda)$ .

El algoritmo escalonado funciona siempre que  $tr$  no tome valores negativos. La condición de fila en  $tr$  no es necesaria pues la función que se utiliza es  $\bar{c}$  que por ser constante en el tiempo sí cumple la condición. Sin embargo el modelo contempla la condición de fila en  $tr$  como parte de la representación teórica por lo que la reformulamos en términos de las flechas de  $\mathfrak{E}(G, tr, \Lambda)$  como sigue: dada una flecha  $(u, v) \in F(G)$  y dos flechas relativas a ella  $((u, t), (v, t^*)), ((u, s), (v, s^*)) \in F(\mathfrak{E}(G, tr, \Lambda))$  de forma que  $t < s$ , se cumple que  $t^* \leq s^*$ .

Denotamos  $\bar{v}_t$  al vértice  $(v, t)$  de la gráfica escalonada  $\mathfrak{E}(G, tr, \Lambda)$ . Del mismo modo definimos:  $\bar{d}(\bar{v}_t)$ ,  $\bar{e}(\bar{v}_t)$  y  $\bar{A}(\bar{v}_t)$  la distancia de la raíz al vértice  $\bar{v}_t$ , la etiqueta del vértice  $\bar{v}_t$  y el conjunto de antecesores del vértice  $\bar{v}_t$  respectivamente. Dicho lo anterior, el *algoritmo escalonado de Dijkstra* es:

### 1. Inicialización

- a)  $\bar{d}(\bar{v}_t) = \infty \forall v \neq x, \forall t \in \{t_0, \dots, \Lambda\}$
- b)
  - $\bar{d}(\bar{x}_0) = t_0$
  - $\bar{d}(\bar{x}_t) = 0 \forall t \in \{t_0 + 1, \dots, \Lambda\}$
- c)  $\bar{e}(\bar{v}_t) = 0 \forall v \neq x, \forall t \in \{t_0, \dots, \Lambda\}$
- d)  $\bar{e}(\bar{x}_0) = 1$
- e)  $\bar{A}(\bar{v}_t) = \emptyset \forall v \in V(G), \forall t \in \{t_0 + 1, \dots, \Lambda\}$
- f)  $Prov = V(\mathfrak{E}(G, tr, \Lambda)) \setminus \{(x, t) | t \in \{t_0 + 1, \dots, \Lambda\}\}$
- g)  $\bar{z}_s = \bar{x}_0$

### 2. Paso algorítmico

- a) Para todo  $\bar{v}_t \in \Gamma^+(\bar{z}_s) = \{\bar{v}_t | \exists a \in \Gamma^+(z), a = (z, v), t = s + tr(a)\}$  establecer:
 
$$\bar{d}(\bar{v}_t) = \begin{cases} \bar{d}(\bar{v}_t) & \text{si } \bar{d}(\bar{v}_t) < \bar{d}(\bar{z}_s) + c(a, s) \\ \bar{d}(\bar{z}_s) + c(a, s) \text{ y } \bar{A}(\bar{v}_t) = \{\bar{z}\} & \text{si } \bar{d}(\bar{v}_t) > \bar{d}(\bar{z}_s) + c(a, s) \\ \bar{d}(\bar{v}_t) \text{ y } \bar{A}(\bar{v}_t) = A(\bar{v}) \cup \{\bar{z}_s\} & \text{si } \bar{d}(\bar{v}_t) = \bar{d}(\bar{z}_s) + c(a, s) \end{cases}$$
 donde  $a = (z, v) \in F(G)$
- b) Encontrar  $\bar{z}_{s^*} \in V(\mathfrak{E}(G, tr, \Lambda))$  tal que  $\bar{d}(\bar{z}_{s^*}) = \min\{\bar{d}(\bar{v}_t) | e(\bar{v}_t) = 0, \bar{v}_t \in V(\mathfrak{E}(G, tr, \Lambda))\}$
- c) Si  $\bar{d}(\bar{z}_{s^*}) < \infty$ , establecer  $\bar{e}(\bar{z}_{s^*}) = 1$ ,  $Prov = Prov \setminus \{\bar{z}_{s^*}\}$
- d) Establecer  $\bar{z}_s = \bar{z}_{s^*}$

### 3. Término

- a) Si  $Prov = \emptyset$       terminar
- b) Si  $\bar{d}(\bar{z}_s) = \infty$       terminar
- c) En otro caso      repetir el paso algorítmico

Ya que esencialmente estamos utilizando el algoritmo clásico, la prueba de que Dijkstra escalonado funciona recae en la de la versión clásica.

### A.3.2. Distancia relativa $\widehat{d}_y^\tau$

Dado  $\tau \in \mathbb{J}$  y  $s \in \Lambda$ , utilizando Dijkstra escalonado sobre la *gráfica volteada*  $\widetilde{\mathfrak{E}}(G, tr, \Lambda)$  con la función volteada de costo  $\widetilde{c}r$  y vértice raíz  $(y_\tau, \widetilde{s})$ , se obtienen las *tr*-rutas de costo mínimo de  $v$  hacia  $y_\tau$  que llegan a él en el tiempo  $s$ . Definimos como  $\widetilde{c}(\widetilde{v}_t)$  y  $\widetilde{d}(\widetilde{v}_t)$  al costo y tiempo de recorrido de las rutas mínimas que inician en  $v$  en el tiempo  $t$  y terminan en  $y_\tau$  en el tiempo  $s$ . Observemos que se cumple  $\widetilde{d}(\widetilde{v}_t) = s - t$ .

Para cada  $\tau \in \mathbb{J}$  definimos la distancia preliminar<sup>11</sup>  $\widehat{\mathbf{d}}[s]: V(G) \times \Lambda \rightarrow \mathbb{N}$  respecto al tiempo de llegada  $s$  como:

$$\widehat{\mathbf{d}}[s](v, t) = \widetilde{c}(v_t)$$

Considerando las distancias preliminares de todos los tiempos de llegada  $s \in \Lambda$  obtenemos el siguiente resultado:

$$\widehat{d}_y^\tau(v, t) = \min\{s \in \Lambda \mid \widehat{\mathbf{d}}[s](v, t)\} \quad (\text{A.6})$$

Este resultado se sigue de que toda *tr*-ruta en  $G$  es un camino en  $\mathfrak{E}(G, tr, \Lambda)$  del mismo costo, y viceversa, por lo que una *tr*-ruta en  $G$  es mínima si y sólo si es mínima como camino en  $\mathfrak{E}(G, tr, \Lambda)$ .

---

<sup>11</sup>Distancia en términos del costo  $\widetilde{c}$ .





# Glosario de Símbolos

## 3.1

Símbolo	Nombre	Página
$\mathfrak{I}$	Conjunto de tipos de usuario o conductores	17
$\tau$	Tipo de usuario (usualmente)	17
$x$	Vértice origen (usualmente)	17
$y$	Vértice destino (usualmente)	17
$\gamma$	Tipo de vehículo (usualmente)	17
$\omega$	Preferencia de usuario (usualmente)	17
$V_x$	Conjunto de orígenes	17
$V_y$	Conjunto de destinos	17
$\Gamma$	Conjunto de tipos de vehículo	17
$\Omega$	Conjunto de preferencias de usuario	17
$D$	Matriz de demanda	18
$\{D_{ijk}\}_{(i,\gamma_j,\omega_k)\in\Lambda\times\Gamma\times\bar{\Omega}}$	familia de matrices de demanda	18
$\Lambda$	Lapso	18
$\pi$	Paso	18
$G$	Gráfica (usualmente)	18
$V(G)$	Vértices de $G$	18
$F(G)$	Flechas de $G$	18
$\mathfrak{T}$	Conjunto de tipos de vía	19
$\lambda(a)$	Longitud	19
$\kappa(a)$	Número de carriles	19
$\kappa(a, b)$	Carriles por operación	19
$v(a)$	Velocidad de proyecto	20
$\varpi(a)$	Tiempo máx de recorrido continuo	20 y 27
$ac(a)$	Ancho de carril	20
$av(a)$	Alineación vertical	20
$csr(a)$	Condiciones de superficie de rodamiento	20
$eu(a)$	Entorno urbano	20
$\vartheta(a)$	Tipo de vía	20
$N^+(v)$	Exvecindad	21
$N^-(v)$	Invecindad	21
$\phi_v(a, b)$	Preferencia de paso	21
$\varsigma(\gamma)$	Tamaño de vehículo	21
$\theta(\gamma)$	Conjunto de vías aptas	21
$\theta^*(\gamma)$	Conjunto adicional de vías	22

$Pref$	Conjunto de preferencias de usuario	22
$pvp$	Preferencia por vías principales	22
$psr$	Preferencia por buena superficie de rodamiento	22
$peu$	Preferencia por entorno urbano	22
$P_I$	Preferencia por vías ideales	22
$F(a, t)$	Flujo general	22
$F_\tau(a, t)$	Flujo parcial	23
$G_\gamma$	Subdigráfica de vías aptas	24
$G_\tau$	Subdigráfica acotada	24
$\sigma$	Radio de vecindad	23
$d_x^\pi(v)$	Distancia a flujo libre desde el origen	23
$d_y^\pi(v)$	Distancia a flujo libre hacia el destino	23
$\Theta_\gamma$	Conjunto general de vías aptas	23
$H_\tau$	Vecindad	24
$\Xi$	Conjunto de flechas auxiliares	24
$G_\tau^*$	Capa	24
$G^*$	Digráfica extendida	24

## 3.2

$J_n$	Jornada	25
$F$	Función de iteración	26
$\mathfrak{S}$	Super-función de iteración	26

## 3.3

$M_\tau(a, t)$	Monitoreo de flecha	27
$(m, \delta)$	Grupo de vehículos en recorrido continuo (usualmente)	27
$\mu_\tau(a, t)$	Vehículos monitoreados de tipo $\tau$	28
$\mu(a, t)$	Vehículos monitoreados	28
$\lambda_M(a, t)$	Espacio ocupado en recorrido continuo	28
$\zeta(a, t)$	Monitoreo de cola	28
$(m, b, \tau)$	Grupo de vehículos en cola (usualmente)	28
$\Upsilon$	Vehículos en cola	28
$\lambda_\zeta(a, t)$	Espacio ocupado en cola	28
$\lambda_\emptyset(a, t)$	Espacio libre	29
$TRC(a, t)$	Tiempo de recorrido continuo	29
$\chi(a, t)$	Demora por cola	29
$TR(a, t)$	Tiempo de recorrido	30
$\rho$	Congestión	30
$D_\lambda(b, t)$	Demanda de espacio	30
$\iota(a, t)$	intencidad de paso	31
$d_y^\tau(v, t)$	Distancia dinámica	31
$P_\tau(a)$	Preferencia global	32
$\widehat{TR}_\tau(a, t)$	Tiempo de recorrido relativo	32
$\widehat{d}_y^\tau(v, t)$	Distancia relativa	32 y 51
$MRC(a, t)$	Memoria de recorrido continuo	33
$M\chi(a, t)$	Memoria de demora	33
$M\lambda_\emptyset(a, t)$	Memoria de espacio libre	33
$MTR(a, t)$	Memoria de tiempo de recorrido	33
$S_\tau(v, t)$	Situación	33
$S_\tau^*(w, b, t)$	Situación futura	34
$E_\tau(b, t)$	Efectividad	34
$\Psi$	Parámetro aleatorio de comportamiento	34 y 55
$E_\tau^\Psi(b, t)$	Efectividad percibida	35
$A_\tau(v, t)$	Función de asignación	35
$I_\tau(a, t)$	Inyección	36

**3.4**

$MF(a, t, n)$	Memoria de flujo	37
$ES(n)$	Estabilidad	37
$N$	Magnitud de muestra	37
$\epsilon$	Parámetro de estabilidad	37

**3.6**

$\widehat{G}_\tau$	Gráfica relativa	51
--------------------	------------------	----

**4**

$\mathcal{O}$	Orden de ejecución	66
---------------	--------------------	----

**A**

$\mathfrak{E}$	Gráfica escalonada	127
$\bar{v}_t$	Vértice escalonado	127
$\bar{d}$	Costo desde la raíz	127

# Bibliografía

- [MCC] “Manual de capacidad de carreteras”. Asociación técnica de carreteras, comité español de la A.I.P.C.R. (Traducción de “Highway Capacity Manual 1985,” Transportation Research Board).
- [OW] Ortuzar, Juan de Dios, and Luis G. Willumsen, “Modeling Transport”. 3rd Ed., John Wiley and Sons Ltd, 2001.
- [WS] Oppenheim, Norbert. “Urban travel demand modeling”. John Wiley and Sons Ltd, 1995.
- [MM] Manheim, Marvin L., “Fundamentals of Transportation Systems Analysis”. The MIT Press, Cambridge, Massachusetts, 1979.
- [CA] Caballero Caballero, Acrisio. Tesis de Licenciatura: “Descripción aplicación e implementación de un modelo de extensión de flujos en redes”. Facultad de Ciencias, UNAM. 2004.
- [CT] Cormen, Thomas H., et all, “Introduction to Algorithms”. The MIT Press, Cambridge, Massachusetts, 1990.
- [CJ] “Instrucciones para subir una escalera”. Julio Cortázar.
- [HC] “The Highway Code”, Gobierno del Reino Unido, <http://www.highwaycode.gov.uk/09.htm#105>.
- [INEGI] <http://www.inegi.gob.mx>
- [W] [http://es.wikipedia.org/wiki/Algoritmo\\_de\\_Dijkstra](http://es.wikipedia.org/wiki/Algoritmo_de_Dijkstra)
- [TM] M.I. Miriam Evelia Télles Ballesteros, Notas del “Curso taller de ingeniería de tránsito”, México 2003.
- [IE] “Inventario de emisiones de la Zona Metropolitana del Valle de México. 2004”.