



ارائه چارچوبی برای تحلیل انطباق نقشه وزن مبنا مبتنی بر الگوی نگاشت کاهش و رایانش ابری

محمد مسعود رحیمی^۱، فرشاد حکیم پور^{۲*}

۱- دانشجوی کارشناسی ارشد سیستم های اطلاعات مکانی، دانشکده مهندسی نقشه برداری و اطلاعات مکانی، پردیس دانشکده های فنی، دانشگاه تهران

۲- استادیار دانشکده مهندسی نقشه برداری و اطلاعات مکانی، پردیس دانشکده های فنی، دانشگاه تهران

چکیده:

این روزها داده های خودروهای شناور (FCD) یکی از موضوعات داغ در زمینه سیستم های حمل و نقل هوشمند (ITS) و مدیریت ترافیک می باشند. با توجه به دقت محدود این داده ها مسئله انطباق نقشه یکی از مسائل داغ در به کارگیری و تحلیل آنی FCD است. در سال های اخیر تحقیقات متعددی بر روی بهبود صحت و نرخ موفقیت این تحلیل انجام پذیرفته است اما موضوع کارآمدی و بهره وری این تحلیل کمتر مورد بررسی قرار گرفته است. با توجه به حجم بالای داده های FCD ذخیره سازی و تحلیل آنی آنان با کمک روش های سنتی امری پیچیده است. زمانی که حجم داده ها از چندین گیگابایت فراتر رود، پایگاه داده های رایطه ای سنتی یا نمی توانند چنین حجمی از داده را پشتیبانی کنند و یا در مدیریت آن دچار مشکلاتی می شوند. به علاوه پردازش حجم بالایی از داده ها نیازمند منابع قدرتمند پردازشی است که این امر نیز نیازمند هزینه های گراف جهت خرید و نگهداری زیرساخت های حجیم رایانشی است. در این مقاله برای پاسخ به چالش های فوق چارچوبی مبتنی بر اصول رایانش ابری و الگوی نگاشت کاهش ارائه شده است. چارچوب فوق تحلیلی توزیع یافته، مقیاس پذیر و کارآمد را بر روی سخت افزارهای معمولی ممکن می سازد. همچنین در این مقاله عملکرد چارچوب بر روی سکوی آپاچی هadoop (Apache Hadoop) مورد ارزیابی های اولیه قرار گرفته است.

واژه های کلیدی: انطباق نقشه وزن مبنا، نگاشت کاهش، هadoop، رایانش ابری.

۱- مقدمه

کاربردهای مدیریت ترافیک مانند پیشنهاد مسیر و تحلیل و نظارت بر جریان ترافیک نیازمند مجموعه داده‌های دقیق و کافی از داده‌های خط سیر خودروها^۱ است [۱]. داده‌های خودروهای شناور^۲ یکی از عناصر اصلی در طیف وسیعی از کاربردها مانند مدیریت و کنترل ترافیک است. داده‌های خودروهای شناور جریانی از نقاط زمانی مکانی هستند که خط سیر یک وسیله نقلیه متحرک را به تصویر می‌کشند [۲]. داده‌های خام خودروهای شناور دارای دقت محدود و نویز فراوان هستند. علت این خطای عمده ایجاد شده توسط ماهواره‌ها و گیرنده‌های سیستم موقعیت یاب جهانی^۳ (GPS) است. همچنین به علت خطای موجود در تولید نقشه و رقومی سازی آن، نقشه‌های مکانی شبکه راهها نیز عاری از خطای نیستند. حاصل این خطاهای آن است که موقعیت حقیقی خودرو بر شبکه مکانی راهها منطبق نمی‌شود؛ حال آنکه می‌دانیم این خودرو بر روی شبکه راهها تردد می‌کرده است. به این پدیده عدم انطباق مکانی گفته می‌شود [۳]. از آنجا که در کاربردهای ITS معمولاً باید مسیر سفر وسیله نقلیه مشخص باشد، داده موقعیت خودرو شناور باید بر روی شبکه راهها منطبق شود که این امر توسط فرآیند پیش پردازشی به نام انطباق نقشه^۴ (MM) انجام می‌پذیرد. هدف از MM یکپارچه سازی موقعیت مکانی خودرو با شبکه راهها، شناسایی یال حقیقی خودرو و تعیین مکان خودرو بر روی آن یال است [۳]. هاشمی و کریمی [۴] الگوریتم‌های موجود انطباق نقشه را به سه دسته کلی ساده، وزن مبنا و پیشرفت‌هه دسته بندی کرده‌اند. نقاط ضعف الگوریتم‌های هندسی به صورت خلاصه سادگی بیش از اندازه و ناتوانی عملکرد در شبکه‌های راه شهری پیچیده می‌باشد [۵]. الگوریتم‌های وزن مبنا از میانگین وزن دار امتیازات برای اختصاص امتیاز به هر یال کاندیدا و انتخاب یال با بیشترین امتیاز استفاده می‌کنند که این امتیازات از معیارهای مشخصی حاصل می‌شود [۶]. در مقابل، الگوریتم‌های پیشرفت‌ه از مدل‌های ریاضیاتی مانند منطق فازی، مدل مخفی مارکوف و غیره برای اختصاص امتیاز به هر یال کاندیدا و انتخاب بهترین یال استفاده می‌کنند. هاشمی و کریمی [۴] نشان دادند گرچه الگوریتم‌های پیشرفت‌ه دارای نرخ بالاتری از موفقیت نسبت به الگوریتم‌های وزن مبنا هستند، منطق آنها پیچیده تر بوده و لذا دارای عملکرد کندتری می‌باشند. بیشتر تحقیقات اخیر در زمینه انطباق نقشه بر روی بهبود دقت انطباق و افزایش نرخ موفقیت انطباق مرکز هستند. در این میان یکی از فاکتورهای فراموش شده موضوع کارآمدی و بهره وری الگوریتم است. در کاربردهای مدیریت ترافیک الگوریتم انطباق نقشه باید بتواند بلاذرنگ عمل کرده و حجم بالایی از FCD را به صورت آنی بر روی نقشه منطبق کند. از طرفی می‌دانیم زمانی که حجم داده‌ها از چندین گیگابایت فراتر رود، پایگاه داده‌های رابطه‌ای سنتی یا نمی‌توانند چنین حجمی از داده را پشتیبانی کنند و یا در مدیریت آن دچار مشکلاتی می‌شوند [۷]. همچنین پردازش حجم بالایی از داده‌ها نیازمند منابع قدرتمند پردازشی است که این امر نیز نیازمند هزینه‌های گزارفته خرید و نگهداری زیرساخت‌های حجمی رایانشی است. لذا ذخیره سازی، پردازش و تحلیل داده‌های FCD در سکو های سنتی چالشی بزرگ محسوب می‌شود. برای پاسخ به این چالش‌ها فناوری رایانش ابری می‌تواند به کمک مدیریت ترافیک بیاید. پیدایش فناوری رایانش ابری راه حلی سودمند، مقیاس پذیر و به صرفه در پاسخگویی به چالش‌های داده‌های بزرگ در کاربردهای حمل و نقلی و مکانی است. راه حل پیشنهادی این مقاله برای پاسخ به چالش‌های مذکور تقسیم بندهی مجموعه داده بزرگ و معرفی هر قسمت از پردازش به یک رایانه برای افزایش توان رایانشی است. بدین منظور ما از الگوی برنامه نویسی نگاشت کاهش^۵ بر روی سکوی هدوپ^۶ استفاده کرده‌ایم. نگاشت کاهش [۸] مدل نوینی از برنامه نویسی توزیع یافته است که اجرای موازی و کارآمد برنامه‌ها را بر روی داده‌های بزرگ و روی سخت افزارهای معمولی ممکن می‌سازد. نگاشت کاهش محاسبات را به دو عملیات کلیدی دسته بندی می‌کند: تابع نگاشت مسئول تقسیم مجموعه داده بزرگ

¹ Car trajectory data

² Floating Car Data (FCD)

³ Global Positioning System (GPS)

⁴ Map Matching (MM)

⁵ MapReduce

⁶ Apache Hadoop

به زیر مجموعه‌ها و پردازش هر کدام از آنها به صورت مستقل است. تابع کاهش نتایج حاصل از نگاشت را تجمیع و ادغام می‌کند [۸]. محبوب ترین پیاده سازی نگاشت کاهش توسط پروژه هدوپ متعلق به بنیاد نرم‌افزاری آپاچی ارائه شده است که یک سکوی محبوب برای رایانش ابری بر روی سخت افزارهای معمولی است [۹]. توانمندی‌های هدوپ به الگوریتم انطباق نقشه امکاناتی مانند پردازش سریعتر، تحمل خطا و تعادل بخشی خودکار بار را هدیه می‌کند. در این پژوهش ما چارچوبی نوین برای انطباق نقشه موازی بر پایه الگوی نگاشت کاهش توسعه داده و عملکرد اولیه آن را بر روی سکوی هدوپ ارزیابی نموده ایم.

۲- تحقیقات مرتبط

در سال‌های اخیر تحقیقات متعددی برای بهبود دقت و عملکرد الگوریتم انطباق نقشه انجام شده است. هاشمی و کریمی [۵] یک الگوریتم وزن مبنا برای موقعیت یابی خودرو در شبکه‌های شهری پیچیده ارائه داده اند. هدف اصلی در این تحقیق ارائه یک الگوریتم کارآمد با حفظ صحت و نرخ موفقیت الگوریتم است. آنها از سه معیار فاصله میان نقطه GPS و قطعه راه، اختلاف میان راستای نقطه GPS و مسیر و اختلاف میان راستای نقاط متواالی GPS و راستای قطعات راه برای شناسایی بهترین قطعه راه در میان سایر کاندیداها استفاده کرده اند. مهمترین خاصیت این الگوریتم نرخ بالای تشخیص صحیح قطعه راه در مناطق شهری است. لیو و همکاران [۱۰] با ارائه یک روش انطباق نقشه وزن دار در کنار یک روش درون یابی بر پایه الگوریتم انطباق، به دقت و کارآمدی مناسبی دست یافتند. آنها از اطلاعات هندسی و توپولوژیک شبکه راهها و بهره‌گیری از اطلاعات تاریخی خط سیر خودروها برای بهبود دقت روش خود بهره برده اند. روش آنان الهام بخش تحقیقات متعددی در سال‌های اخیر شده است. خیمنز و همکاران [۱۱] یک الگوریتم انطباق نقشه برای مناطق شهری با کیفیت پایین سیگنال سامانه ماهواره‌ای ناوبری جهانی^۱ (GNSS) ارائه نموده اند. هدف اصلی این تحقیق ارائه روشی برای حل خطا در موقعیت یابی و انطباق نقشه است، زمانی که دریافت سیگنال GNSS با مشکل روبرو است و میزان خطای موقعیت یابی بالا است. قدوس و واشنگتن [۱۲] از الگوریتم کوتاه ترین مسیر^{*} A و داده خط سیر برای افزایش دقت فرآیند انطباق نقشه در داده‌های GPS با نرخ نمونه‌برداری پایین استفاده کرده اند. ^{*}A برای تشخیص کوتاه ترین مسیر میان دو نقطه متواالی GPS استفاده شده است. در این محاسبه پارامترهای توپولوژیک مثل اتصال و محدودیت دور زدن نیز لحاظ شده است. چن و همکاران [۱۳] نیز برای حل چالش داده‌های با نرخ نمونه برداری پایین الگوریتمی با برنامه نویسی چندمعیاره پویا^۲ (MDP) ارائه داده اند. در این الگوریتم MDP برای کاهش تعداد یال‌های کاندیدا و افزایش دقت انتخاب بهترین یال استفاده شده است. با هدف بهبود عملکرد و کارایی الگوریتم‌های انطباق نقشه، لی و همکاران [۱۴] الگوریتمی موثر به نام ^{*}Passby را ارائه کرده اند. این الگوریتم به جای انطباق تمامی نقاط GPS تنها آنهایی را که به تقاطعات نزدیک هستند شناسایی و منطبق می‌کند. بدین ترتیب حجم پردازشی کاهش یافته و از پردازش نقاط اضافی میانی جلوگیری به عمل می‌آید. به همین ترتیب هوانگ و همکاران [۱۵] الگوریتمی نسبتا مشابه با ^{*}Passby توسعه داده اند. در این تحقیق روشی برای بهبود نحوه انتخاب نقاط میانی در الگوریتم ^{*}Passby ارائه شده است. هوانگ و همکاران جزو اندک محققین زمینه انطباق نقشه هستند که از سکوی هدوپ و رایانش ابری برای بهبود کارایی و عملکرد الگوریتم انطباق نقشه استفاده نموده‌اند.

¹ Global Navigation Satellite System

² Multi-Criteria Dynamic Programming (MDP)

۳- چارچوب پیشنهادی انطباق نقشه وزندار مبتنی بر الگوی نگاشت کاوش و هدوپ

ایده اصلی در این روش استفاده از نقشه شبکه راهها در کتاب داده های زمانی مکانی خودروهای شناور است. دو گام اصلی در این روش مرحله پیش پردازش و مرحله انطباق نقشه است که در ادامه به تفصیل بررسی خواهد شد.

۳-۱- ذخیره سازی داده های خودروهای شناور و داده های شبکه راهها

به صورت کلی، دو دسته منبع داده در چارچوب طراحی شده وجود دارد. دسته اول نقشه شبکه راهها است که خودرو ببروی آن در حال حرکت است و دسته دوم جریان پیوسته ای از داده های خودروهای شناور است که برای پردازش های آتی در قالب داده های تاریخی در سیستم ذخیره می شود. داده های خودروهای شناور توسط تجهیزات مکانیاب خودرو^۱ (AVL) اندازه گیری شده و توسط فناوری های ارتباطی مانند سیستم های ارتباطی سلولی (3G/4G) به سمت کارساز^۲ مرکزی تعیین شده ارسال می شوند. کارساز به کمک عامل فلوم^۳ آپاچی داده ها را دریافت کرده و در فایل سیستم توزیع یافته هدوپ^۴ (HDFS) ذخیره می کند. HDFS یک سیستم ذخیره سازی توزیع یافته برای ذخیره سازی و انتقال مطمئن حجم بسیار بالایی از داده های ساختار یافته یا غیر ساختار یافته بر روی خوشه ها می باشد [۹]. شبکه راهها نیز گرافی جهت دار G(V,E) است که در آن ۷ مجموعه ای از گره ها است که بیانگر نقاط اصلی و نقاط تقاطع راهها می باشد و E مجموعه ای یال ها است که بیانگر قطعاتی از راه است که دو گره را بهم پیوند می دهد. در شبکه راهها هر یال دارای اطلاعاتی مانند گره شروع، گره پایان و توضیحات است. ما در این چارچوب شبکه راهها را در یک پایگاه داده غیر رابطه ای به نام اج بیس^۵ ذخیره کرده ایم که برروی HDFS بنا می شود. Hadoop از معماری ارباب و برد^۶ بهره می برد. گره نام^۷ و دنبال کننده کار^۸ گره های ارباب هستند حال آنکه گره داده^۹ و دنبال کننده وظیفه^{۱۰} گره های برد در خوشه هستند. داده ورودی به بلوك ها تقسیم بندی شده تقسیم بندی شده و اطلاعات این بلوك ها در گره نام قرار می گیرند که ابر داده^{۱۱} هر بلوك را حفظ می کند، لذا سیستم Hadoop می داند کدام گره داده ذخیره شده است. اگر یک گره چار مشکل شود تمامی کار از بین نمی بود چرا که Hadoop می داند نسخه المثلی آن بلوك ها را کجا ذخیره کرده است [۱۶]. دنبال کننده کار و دنبال کننده وظیفه نظارت بر اجرای فرآیند ها را بر عهده دارند. آنها رابطه مشابهی با گره نام و گره داده دارند. دنبال کننده وظیفه مسئول اجرای وظیفه و ارسال پیام ها به دنبال کننده کار است. دنبال کننده کار با دنبال کننده وظیفه در ارتباط است و بر اجرای فرآیند نظارت دارد. اگر دنبال کننده کار متوجه شود که یک دنبال کننده وظیفه شکست خورده است و یا در اتمام بخشی از وظایف خود ناتوان است، وظیفه آن را به دنبال کننده وظیفه دیگری می سپارد [۹].

¹ Automatic Vehicle Location (AVL)

² Server

³ Apache Flume

⁴ Hadoop Distributed File System (HDFS)

⁵ HBase

⁶ Master-slave

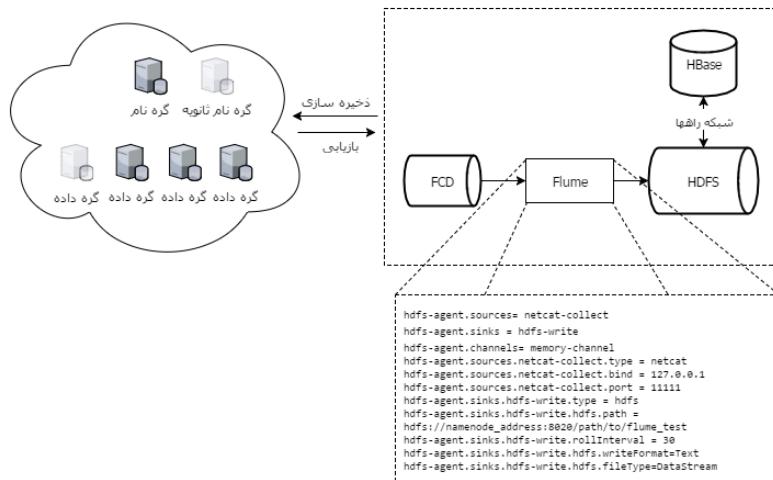
⁷ Name Node

⁸ Job Tracker

⁹ Data Node

¹⁰ Task Tracker

¹¹ Metadata



شکل ۱: ذخیره‌سازی داده‌های خودروهای شناور و داده‌های شبکه راهها در چارچوب پیشنهادی

۲-۳- پیش پردازش داده‌ها و شاخص‌بندی مکانی

مرحله پیش پردازش به دو بخش پیش پردازش نقشه راهها و پیش پردازش نقاط GPS تقسیم بندی می‌شود. با هدف ذخیره سازی و دسترسی موثر به داده‌های مکانی در HDFS، نیاز به شاخص بندی مکانی احساس می‌شود. اساساً شاخص بندی مکانی با هدف بهبود عملکرد پرس و جو‌های مکانی ایجاد می‌شود. تا به امروز با هدف سرعت بخشیدن به فرآیند پرس و جوی مکانی، روش‌های متعددی برای ساختار بندی شاخص طراحی و ارائه شده است که از جمله آنها می‌توان به شاخص شبکه‌ای، درخت آر^۱ [۱۷] و چار درخت [۱۸]^۲ اشاره نمود. با این حال انتقال ساختار این شاخص‌ها به سیستم‌های دیگر با رویکرد داده‌های بزرگ چالش متفاوتی است. یکی از محدودیت‌های سیستم هدوب آن است که در سیستم توزیع یافته فایل هدوب (HDFS)، فایل تنها یکبار نوشته شده و بارها خوانده می‌شود. این بدین معنی است که ویرایش فایل عملاً ممکن نیست که این محدودیتی بزرگ در مقایسه با فایل سیستم‌های سنتی است [۱۹]. با توجه به محدودیت فوق و ماهیت پویای داده‌های خودروهای شناور، در این چارچوب شاخص مکانی برای شبکه راهها که ماهیتی ایستاده ایجاد می‌شود. این شاخص می‌تواند سلول‌بندی منظم با تقسیم منطقه جغرافیایی به k منطقه کوچکتر با ابعاد برابر باشد. بدین ترتیب می‌توان سلول‌ها را بر اساس تعداد گره‌های پردازشی موجود در خوش هدوب به گره‌های مختلف اختصاص داد. لذا ابتدا شبکه راهها به سلول‌هایی تقسیم بندی شده و هر یال از شبکه به این سلول‌ها اختصاص می‌یابد. به هر سلول یک شناسه یکتا اختصاص داده می‌شود و برای آن لیست تمامی یال‌هایی که درون آن واقع شده‌اند و یا از آن گذشته‌اند ذخیره می‌شود. پیش پردازش داده‌های خودروی شناور نیز شامل حذف داده خودروهایی است که دارای قرائت‌های خارج از حدود شبکه راهها هستند و یا نرخ نمونه برداری آنها آنچنان قلیل بوده که فاصله زمانی میان دو گزارش متواالی از حد آستانه مشخصی بیشتر باشد. [۱۰] این حد آستانه را به صورت تجربی ۲۰ دقیقه در نظر گرفته است.

۳-۳- انطباق نقشه مبتنی بر الگوی نگاشت کاهش

در مدل پیشنهادی برای انطباق نقشه توزیع یافته، ما از دو برنامه نگاشت کاهش استفاده می‌کنیم. برنامه نخست (Job1) شامل شناسایی منطقه کاندیدا، شناسایی یال‌های کاندیدا و وزندهی به یال‌های کاندیدا است و برنامه دوم (Job2) شامل تعیین یال حقیقی خودرو و تعیین مکان خودرو در شبکه راهها است. در Job1 نیاز است

¹ R-Tree

² Quad-Tree

منطقه کاندیدا شناسایی شود. اینکار توسط تابع نگاشت و در قالب رابطه ۱ انجام می‌پذیرد. به تابع نگاشت کلیدهای ورودی معرفی شده و برای هر کلید پردازنه‌ای اختصاص داده شده تا تمام مقادیر مورد پردازش قرار گیرند.

$$\text{Map}(PID, Pinfo) \rightarrow (PID, \text{list}(GridID)) \quad \text{رابطه (۱)}$$

سپس با کمک تابع بر زننده^۱ داده‌ها با کلیدهای یکسان تجمعی می‌شود ($\text{list}(PID, \text{list}(GridID))$). در مرحله شناسایی یال‌های کاندیدا، یال‌های درون مناطق مرتبط با هر کلید شناسایی شده و فاصله میان یال‌ها تا نقطه GPS و همچنین اختلاف راستای میان خودرو و یال اندازه‌گیری می‌شود. برای محاسبه مشابهت میان راستای خودرو و یال ذکر این نکته ضروری است که در زمان حرکت خودرو با سرعت پایین، خطای موجود در راستای خودرو بالا است. لذا زمانی باید از قید شباهت میان راستا استفاده شود که میزان سرعت خودرو از حد آستانه مشخصی بیشتر باشد. [۱۰] این حدآستانه را به صورت تجربی ۱۰.۸ کیلومتر بر ساعت در نظر گرفته است. بعد از شناسایی مجموعه یال‌های کاندیدای هر نقطه، ما برای هر لینک بر پایه سه فاکتور وزن دهی انجام می‌دهیم. این سه فاکتور عبارتند از نزدیکی میان نقطه GPS و یال، شباهت میان راستای خودرو و راستای لینک و اتصال یال‌ها. تابع وزن دهی برای دو فاکتور اول بر پایه تحلیل‌های هندسی توسعه یافته است در حالی که فاکتور اتصال یال بر پایه تحلیل توبولوژیک توسعه یافته است. قید نزدیکی بر پایه فاصله میان نقطه GPS و یال ایجاد می‌شود. هر چه فاصله میان نقطه و یال کوچکتر باشد، احتمال اینکه یال همان یال حقیقی باشد بیشتر است. لذا در این روش تابع وزن برای نزدیکی در قالب رابطه ۲ تعیین می‌شود:

$$WS_D = W_D * (1 - 0.5 * \frac{d_i + d_{i+1}}{\text{MaxDistanceError}}) \quad \text{رابطه (۲)}$$

در رابطه (۲)، فاصله میان نقطه فعلی GPS به نام P_i تا لینک کاندیدای خود، d_{i+1} فاصله میان نقطه GPS بعدی P_{i+1} تا لینک کاندیدای خود، W_D ضریب تاثیر وزن نزدیکی و MaxDistanceError حداکثر مقدار خطای مکانی است که ۱۰۰ متر در نظر گرفته می‌شود. قید راستا بر اساس اختلاف راستای میان خودرو و یال تعیین می‌شود. راستای خودرو به شدت به راستای یال وابسته است. در روش پیشنهادی وزن راستا به کمک رابطه ۳ تعیین می‌شود:

$$WS_H = W_H * (1 - 0.5 * \frac{\text{angle}_i + \text{angle}_{i+1}}{\text{MaxAngle}}) \quad \text{رابطه (۳)}$$

در رابطه (۳)، angle_i اختلاف میان راستای خودرو و یال برای نقطه فعلی GPS به نام P_i ، P_{i+1} اختلاف میان راستای خودرو و یال برای نقطه بعدی GPS به نام P_{i+1} ، W_H ضریب تاثیر وزن راستا و MaxAngle حداکثر زاویه میان راستای خودرو و راستای یال است که آن را ۶۰ درجه در نظر می‌گیریم. دو یال به صورت مستقیم متصل اند اگر و تنها اگر نقطه ابتدای یکی، نقطه انتهای دیگری باشد. در وزن دهی بر اساس اتصال یال‌ها، ما دو نقطه قبل و بعد از نقطه فعلی GPS را لحاظ می‌کنیم. در این روش وزن اتصال از طریق رابطه ۴ و به صورت زیر محاسبه می‌شود:

$$WS_{LC} = W_{LC} * \frac{w_i + w_{i+1}}{2} \quad \text{رابطه (۴)}$$

در رابطه ۴، W_{LC} ضریب تاثیر وزن اتصال یال‌هاست. w_i و w_{i+1} متغیرهایی است که بیانگر اتصال میان نقاط P_i و P_{i+1} و اتصال میان P_i و P_{i+1} هستند. به صورت تجربی ما ضرایب وزن را برابر فرض می‌کنیم. در مدل پیشنهادی وزن نهایی از طریق مجموع سه عامل وزن بالا تعیین می‌شود. بدین ترتیب همانند رابطه ۵، در تابع کاهش برای هر کلید، لیستی از یال‌ها و وزن‌های مربوط به هر کدام ایجاد می‌شود.

¹ Shuffle

$$Reduce(PID, list(GridID)) \rightarrow (PID, list(LinkID, W)) \quad \text{رابطه (۵)}$$

در نهایت سیستم هوشمند هدوب، تمامی خروجی های حاصل ازتابع کاهش را جمع بندی کرده و آنها را بر حسب PID دسته بندی می کند تا خروجی نهایی $Job1$ حاصل شود ($list(PID, list(LinkID, W))$).

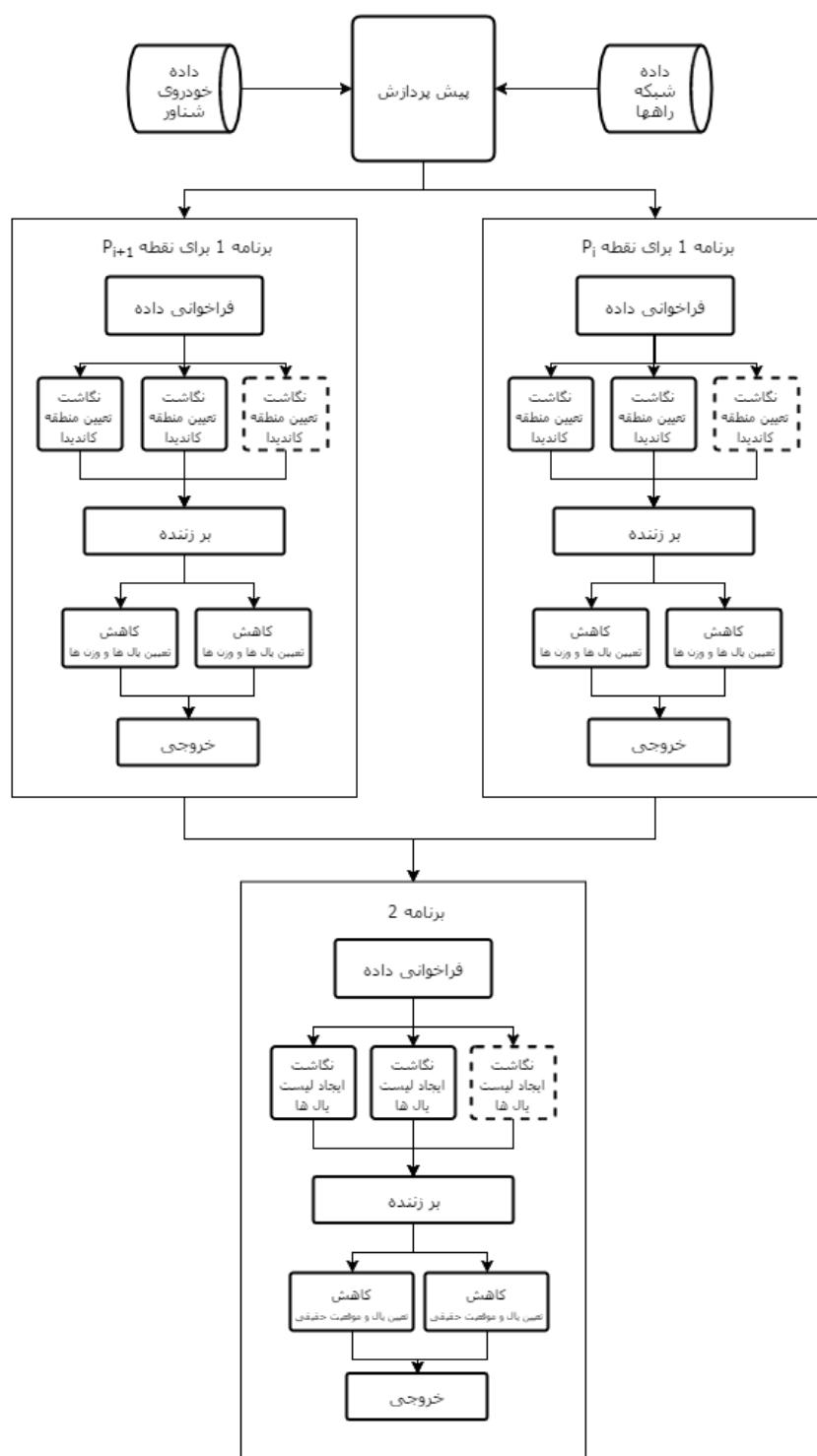
$Job2$ شامل تعیین یال حقیقی و نقطه نهایی حاصل از فرآیند انطباق نقشه است. در این گام باید وزن کلی برای هر مسیر کاندیدا از نقطه GPS پیشین P_{i-1} تا نقطه P_{i+1} بعدی محاسبه گردد و به عنوان امتیاز آن مسیر اطلاق گردد. از آنجا که نقطه پیشین بر یال حقیقی خود منطبق شده است، تعداد مسیر های کاندیدا توسط تعداد یال های کاندیدا برای P_i و P_{i+1} تعیین می شود. هر یال کاندیدای متعلق به P_i عضوی از حداقل یک مسیر است. برای هر یال، ما تنها مسیری که دارای بالاترین امتیاز است را ذخیره می کنیم و این امتیاز به عنوان وزن یال تعیین می گردد. یالی که دارای بیشترین امتیاز است به عنوان یال حقیقی انتخاب شده و نزدیکترین نقطه بر روی یال به P_i به عنوان مکان خودرو بر روی یال انتخاب می گردد. بدین منظور ابتدا $Job1$ برای نقطه P_i و P_{i+1} اجرا شده و نتایج حاصل به عنوان ورودی به $Job2$ معرفی می شوند. سپس از میان یال های کاندیدای متعلق به P_i ، آنها که به یال اختصاص یافته شده به P_{i-1} متصل اند شناسایی می شوند. اگر در این لیست یالی وجود داشت که به یال تعیین شده برای P_{i-1} متصل بود، آن را به عنوان یکی از کاندیداهای اصلی به لیست ثانویه اضافه می کنیم. حال برای هر یال در لیست ثانویه، ابتدا یال های بعدی متصل به یال کاندیدای اصلی پیدا شده و با مقایسه با یال های موجود در لیست P_{i+1} ، مسیر های کاندیدا تشکیل می شود. در این روش پیشنهادی، رابطه ۶ بیانگر تابع نگاشت $Job2$ است:

$$Map(P_i, list(LinkID, W)) \rightarrow (LinkID_{P_i}, LinkID_{P_{i+1}}) \quad \text{رابطه (۶)}$$

تابع بروزرنده مقادیر دارای کلید یکسان را تجمعی کرده تا بدین وسیله برای هر یال کاندیدای P_i تمامی یال های بعدی آن که P_{i+1} ممکن است در حقیقت روی آنها قرار داشته باشد تعیین شوند ($LinkID_{P_i}, list(LinkID_{P_{i+1}})$). سپس برای هر P_i مسیری که دارای بالاترین امتیاز است نگهداری می شود. بدین منظور وزن یال های مربوطه جمع شده و مسیری که شامل یال کاندیدا بوده و دارای بیشترین امتیاز است به عنوان مسیر اصلی انتخاب شده و سایر مسیرها حذف می گردند. سپس امتیاز مسیر به عنوان وزن یال کاندیدا برای P_i تعیین می گردد. در نهایت با مقایسه یال های کاندیدا متعلق به P_i یالی که دارای بیشترین وزن است به عنوان یال نهایی انتخاب شده و نزدیکترین نقطه یال به نقطه GPS به عنوان موقعیت حقیقی GPS تعیین می گردد. در این مدل، تابع کاهش $Job2$ توسط رابطه (۷) تعیین می شود:

$$Reduce(LinkID_{P_i}, list(LinkID_{P_{i+1}})) \rightarrow (P_i, P_{MapMatched}) \quad \text{رابطه (۷)}$$

سیستم تمامی خروجی های $Job2$ را تجمعی و فهرستی از نقاط منطبق شده ارائه می کند. نمودار روند الگوریتم توزیع یافته انطباق نقشه در شکل (۲) نمایش داده شده است.



شکل ۲: جریان کار تحلیل انطباق نقشه در چارچوب پیشنهادی

۴- ارزیابی نتایج اولیه

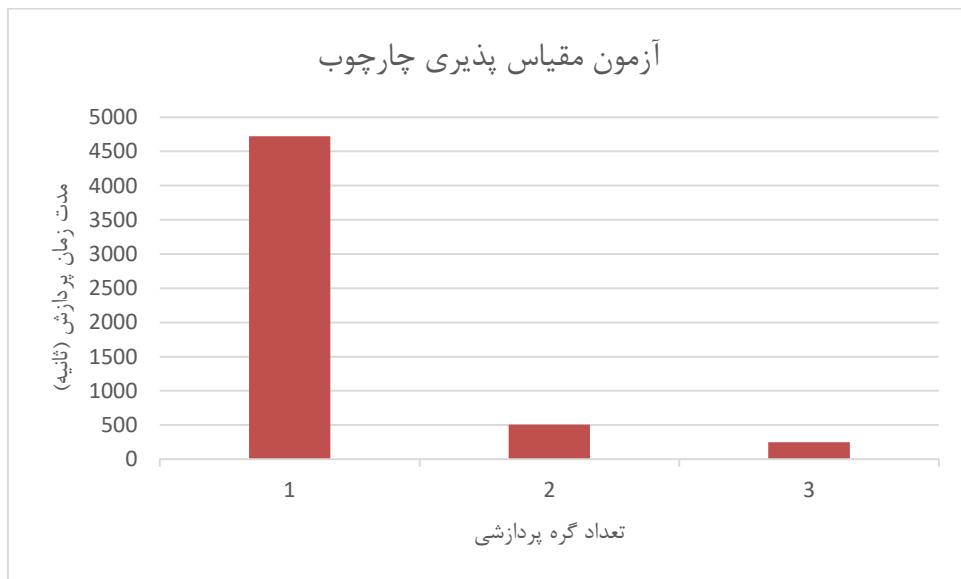
در این بخش یک ارزیابی از نحوه عملکرد چارچوب انطباق نقشه پیشنهادی ارائه می‌گردد. برای ارزیابی از داده‌های واقعی خودروهای شناور آژانس حمل و نقل شهرداری سانفرانسیسکو استفاده نموده‌ایم. بدین‌منظور بیش از چهار میلیون رکورد داده متعلق به یک بازه زمانی یک هفته‌ای با نرخ‌های نمونه برداری متفاوت مورد ارزیابی و

تحلیل قرار گرفت. نمونه رکورد این داده ها در جدول (۱) نمایش داده شده است. برای شبکه راهها نیز داده شبکه راههای *OSM*^۱ به کار گیری شده است.

جدول ۱: نمونه داده خودرو شناور متعلق به آزادس حمل و نقل شهرداری سانفرانسیسکو

زاویه راستا	سرعت خودرو	عرض جغرافیایی	طول جغرافیایی	شناسه خودرو	زمان گزارش
۱۷۰	۳.۸۸	۳۷.۸۰۶۳۳	-۱۲۲.۴۲۰۴۷	۰۱	۲۰۱۶/۰۳/۰۱ ۰۰:۰۳:۲۹

برای ارزیابی نحوه عملکرد چارچوب پیشنهادی ما الگوریتم انطباق نقشه ارائه شده را بروی یک خوشه هدوپ و به زبان جاوا در محیط توسعه *Eclipse* پیاده سازی کرده ایم. گره نام خوشه دارای ۸ گیگابایت *RAM* و پردازنده چهارهسته ای *Intel* و گره های داده دارای چهار گیگابایت *RAM* و پردازنده دو هسته ای *Intel* بودند. همانطور که پیش از این نیز بیان شد هدف اصلی استفاده از الگوی نگاشت کاهش این است که بتوان ذخیره سازی، پردازش و تحلیل های داده های بزرگ را بروی سخت افزار متوسط استفاده شده است. در ارزیابی های اولیه عملکرد چارچوب پیشنهادی برای داده مورد آزمایش بروی تعداد متغیری از گره های پردازشی مورد بررسی قرار گرفت. نتیجه عملکرد چارچوب در شکل (۳) نمایش داده شده است. همانطور که مشاهده می شود افزایش تعداد گره پردازشی با مدت زمان پردازش نسبت عکس دارد.



شکل ۳ مدت زمان پردازش در چارچوب پیشنهادی با افزایش تعداد گره های پردازشی

ما همچنین سرعت انطباق نقشه را با تعداد گره های پردازشی مختلف مورد ارزیابی قرار داده ایم که نتایج بسیار امیدبخشی حاصل شد. شکل ۴ بیانگر عملکرد چارچوب و سرعت انطباق نقشه است. این تحقیق در حال توسعه است و امیدواریم که با پیشرفت های بیشتر در تحقیقات، ارزیابی ها و نتایج بهتری تهیه و ارائه گردد.

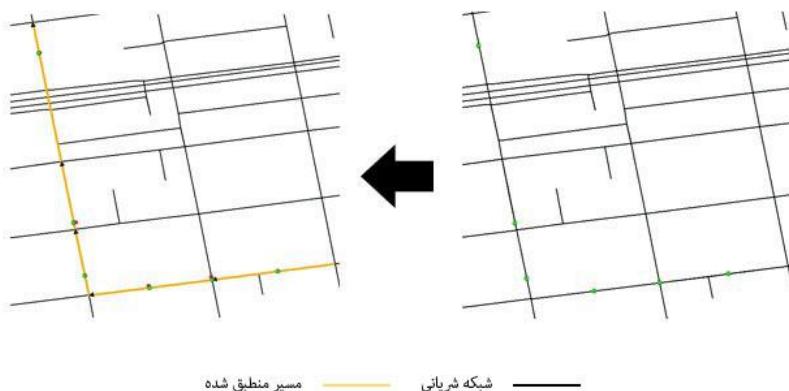
^۱ www.openstreetmap.org

تست سرعت انطباق نقشه در چارچوب



شکل ۴: سرعت انطباق در چارچوب پیشنهادی با افزایش تعداد گره های پردازشی

در نهایت حاصل اجرای تحلیل انطباق نقشه خط سیر هایی به صورت شکل ۵ است.



شکل ۵ خط سیر حاصل از فرآیند انطباق نقشه

۵- نتیجه گیری و تحقیقات آتی

مدیریت کارآمد و بهینه ترافیک نیازمند تحلیل های آنی و لحظه ای داده های حجمی ترافیکی است. یکی از مهمترین داده های حجمی ترافیکی، داده های خودروهای شناور است. از آنجا که در کاربردهای مدیریت ترافیک معمولاً باید مسیر سفر وسیله نقلیه مشخص باشد، داده موقعیت خودرو شناور باید با شبکه راهها منطبق شود که این امر به کمک تحلیل انطباق نقشه صورت می پذیرد. در این مقاله ما یک چارچوب توزیع یافته و مقیاس پذیر برای تحلیل وزن دار انطباق نقشه به صورت موازی ارائه کردیم. این چارچوب که بر مبنای الگوی نگاشت کاهش و برروی سکوی هدوپ توسعه یافته است می تواند برای پردازش داده های آنی و تاریخی خودروهای شناور با حجم بالا به کارگیری شود. توانمندی های هدوپ به الگوریتم انطباق نقشه امکاناتی مانند پردازش سریعتر، تحمل خطا و تعادل بخشی خودکار بار را هدیه می کند. نتایج اولیه حاکی از سرعت بالای الگوریتم ارائه شده برای انطباق نقشه در چارچوب پیشنهادی است. این تحقیق همچنان در حال توسعه است و پیشرفتها و نتایج آتی در قالب گزارشات کاملتر ارائه خواهد شد.

مراجع

- [1] S. Liu, Y. Liu, L. M. Ni, J. Fan, and M. Li, "Towards mobility-based clustering," in Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining, 2010, pp. 919-928.
- [2] M. A. Quddus, W. Y. Ochieng, and R. B. Noland, "Current map-matching algorithms for transport applications: State-of-the art and future research directions," Transportation research part c: Emerging technologies, vol. 15, pp. 312-328, 2007.
- [3] M. A. Quddus, "High integrity map matching algorithms for advanced transport telematics applications," Citeseer, 2006.
- [4] M. Hashemi and H. A. Karimi, "A critical review of real-time map-matching algorithms: Current issues and future directions," Computers, Environment and Urban Systems, vol. 48, pp. 153-165, 2014.
- [5] M. Hashemi and H. A. Karimi, "A weight-based map-matching algorithm for vehicle navigation in complex urban networks," Journal of Intelligent Transportation Systems, pp. 00-00, 2016.
- [6] L. Li, M. Quddus, and L. Zhao, "High accuracy tightly-coupled integrity monitoring algorithm for map-matching," Transportation Research Part C: Emerging Technologies, vol. 36, pp. 13-26, 2013.
- [7] G. Aydin, I. R. Hallac, and B. Karakus, "Architecture and implementation of a scalable sensor data storage and analysis system using cloud computing and big data technologies," Journal of Sensors, vol. 2015, 2015.
- [8] J. Dean and S. Ghemawat, "MapReduce: simplified data processing on large clusters," Communications of the ACM, vol. 51, pp. 107-113, 2008.
- [9] T. White, Hadoop: The definitive guide: " O'Reilly Media, Inc.", 2012.
- [10] S. Liu, C. Liu, Q. Luo, L. M. Ni, and R. Krishnan, "Calibrating large scale vehicle trajectory data," in 2012 IEEE 13th International Conference on Mobile Data Management, 2012, pp. 222-231.
- [11] F. Jiménez, S. Monzón, and J. E. Naranjo, "Definition of an Enhanced Map-Matching Algorithm for Urban Environments with Poor GNSS Signal Quality," Sensors, vol. 16, p. 193, 2016.
- [12] M. Quddus and S. Washington, "Shortest path and vehicle trajectory aided map-matching for low frequency GPS data," Transportation Research Part C: Emerging Technologies, vol. 55, pp. 328-339, 2015.
- [13] B. Y. Chen, H. Yuan, Q. Li, W. H. Lam, S.-L. Shaw, and K. Yan, "Map-matching algorithm for large-scale low-frequency floating car data," International Journal of Geographical Information Science, vol. 28, pp. 22-38, 2014.
- [14] Y. Li, C. Liu, K. Liu, J. Xu, F. He, and Z. Ding, "On efficient map-matching according to intersections you pass by," in International Conference on Database and Expert Systems Applications, 2013, pp. 42-56.
- [15] J. Huang, J. Qie, C. Liu, S. Li, J. Weng, and W. Lv, "Cloud computing-based map-matching for transportation data center," Electronic Commerce Research and Applications, vol. 14, pp. 431-443, 2015.
- [16] A. Bialecki, M. Cafarella, D. Cutting, and O. O'MALLEY, "Hadoop: a framework for running applications on large clusters built of commodity hardware," Wiki at <http://lucene.apache.org/hadoop>, vol. 11, 2005.
- [17] A. Guttman, R-trees: a dynamic index structure for spatial searching vol. 14: ACM, 1984.
- [18] R. A. Finkel and J. L. Bentley, "Quad trees a data structure for retrieval on composite keys," Acta informatica, vol. 4, pp. 1-9, 1974.
- [19] A. Eldawy and M. F. Mokbel, "The era of big spatial data," in Data Engineering Workshops (ICDEW), 2015 31st IEEE International Conference on, 2015, pp. 42-49.