دانشگاه صنعتی امیرکبیر

(پلی‌تکنیک تهران)

دانشکده مهندسی کامپیوتر و فناوری اطلاعات

پایان نامه کارشناسی ارشد

گرایش نرم افزار

حل مسائل بهینه‌سازی در شبکه‌های تصادفی با استفاده از اتوماتاهای یادگیر

نگارش: مهدی قربعلی‌پوردرو

استاد راهنما: دکتر محمدرضا میبدی

اسفند ۱۳۸۷

بسمه تعالی

شماره مدرک:

فرم اطلاعات پایان‌نامه
کارشناسی- ارشد و دکترا
کتابخانه مرکزی

مشخصات دانشجو	نام خانوادگی : قربعلی پور درو	نام : مهدی	شماره دانشجویی: ۸۵۱۳۱۰۵۳
	دانشکده : مهندسی کامپیوتر وفناوری اطلاعات	رشته : مهندسی کامپیوتر	گروه : نرم افزار
عنوان	حل مسائل بهینه‌سازی در شبکه‌های تصادفی با استفاده از اتوماتاهای یادگیر		
Title	Solving Optimization Problems in Stochastic Networks Using Learning Automata		

استاد راهنما	نام خانوادگی : میبدی	درجه و رتبه	استاد راهنما	نام خانوادگی :	درجه و رتبه
	نام : محمدرضا	استاد		نام :	
استاد مشاور	نام خانوادگی :	درجه و رتبه	استاد مشاور	نام خانوادگی :	درجه و رتبه
	نام :			نام :	

دانشنامه	○دکترا ●ارشد ○کارشناسی	سال تحصیلی :۸۵-۸۷
نوع پروژه	●بنیادی ○ توسعه ای ○ نظری ●کاربردی	

مشخصات ظاهری	تعداد صفحات ۲۳۴	تصویر ● جدول ● نمودار ● نقشه ○ واژه‌نامه ●	تعداد مراجع ۱۱۶	●ضمائم تعداد صفحات:۴۷
زبان متن	فارسی ● انگلیسی ○	چکیده	فارسی ●	انگلیسی ●
یاداشت				
توصیفگر				
کلید واژه فارسی	مسائل بهینه‌سازی، گراف تصادفی، اتوماتای یادگیر			
Key word of English	Optimization Problems, Stochastic Graph, Learning Automata			

پایان‌نامه **"حل مسائل بهینه‌سازی در شبکه‌های تصادفی با استفاده از اتوماتاهای یادگیر"** در تاریخ ۸۷/۱۲/۱۲ مورد پذیرش هیأت محترم داوران با رتبه و نمره قرار گرفت.

استاد راهنما و رئیس هیأت داوران: دکتر محمدرضا میبدی

داور داخلی: دکتر محمد رحمتی

داور خارجی: دکتر حمید بیگی (دانشگاه صنعتی شریف)

"سر ارادت ما و آستان حضرت دوست که هر چه بر سر ما می رود ارادت اوست"

به یاد پیر مرد مهربان

سید موسی موسوی

و اولین معلم ریاضی ام

سید حسن زمانی

و کلیه بزرگانی که در خاک روستایم در آرامش ابدی خفتند

ابش قربعلی زاده، ابراهیم طاهر پور، خالق اله یاری و . . .

خداوند جمله را بیامرزد و روحشان را شاد کند

خداوند بزرگ را سپاسگزارم که توانستم پایان نامه دوره کارشناسی ارشد خود را زیر نظر جناب آقای دکتر محمدرضا میبدی به پایان برسانم و از راهنماییها و نکتهای علمی سودمند ایشان بهره بگیرم. آرزوی سلامتی و سرافرازی هر چه بیشتر ایشان را از درگاه خداوند متعال خواستارم.

چکیده

یک گراف تصادفی G را توسط سه تایی $G=<V,E,F>$ تعریف می‌کنیم که در آن $V=\{1,2,3,...,n\}$ مجموعه راسهاست و $E\subset V\times V$ مجموعه یالها را معین می‌کند و ماتریس F که یک ماتریس $n\times n$ است مشخص کننده توزیع احتمالی وزن یالهای گراف می‌باشد. در این پایان نامه الگوریتمهایی برای حل چهار مساله در گرافهایی تصادفی در حالتی که توزیع وزن یالهای گراف تصادفی از قبل شناخته شده نیست پیشنهاد شده است. این مسائل عبارتند از: یافتن درخت پوشایی از گراف تصادفی با مینیمم هزینه مورد انتظار، یافتن تطابقی از گراف تصادفی با ماکزیمم وزن مورد انتظار، یافتن درخت کوتاهترین مسیر از یک راس مبدا به سایر گره‌ها با مینیمم طول مورد انتظار و یافتن درخت اشتاینری از گراف تصادفی با مینیمم هزینه مورد انتظار. برای ارزیابی کارایی الگوریتمهای پیشنهادی، این الگوریتمها با الگوریتمهای ارائه شده قبلی در صورت وجود، مقایسه شده اند و درغیر این صورت با روشی به نام نمونه‌گیری استاندارد سنجیده شده اند و نشان داده شده است که تعداد نمونه‌های گرفته شده با توجه به درصد همگرایی الگوریتم به جواب بهینه، در الگوریتمهای پیشنهادی کمتر از الگوریتمهای مشابه می‌باشد.

کلمات کلیدی

مسائل بهینه‌سازی، گراف تصادفی، اتوماتای یادگیر

فهرست مطالب

۱- مقدمه ۲

۱-۱ گراف تصادفی ۲

۱-۱-۱ چه الگوریتمهایی برای گرافهای تصادفی مناسبند؟ ۳

۱-۲ درخت پوشای مینیمم در گرافهای تصادفی ٤

۱-۲-۱ معرفی مسأله درخت پوشای مینیمم ٤

۱-۲-۲ درخت پوشای مینیمم در گرافهای تصادفی ٦

۱-۲-۳ نگارشهای دیگر مسأله ۷

۱-۳ تطابق وزندار ماکزیمم در گرافهای تصادفی ۱۰

۱-۳-۱ معرفی مسأله تطابق وزندار ماکزیمم ۱۰

۱-۳-۲ تطابق وزندار ماکزیمم در گرافهای تصادفی ۱۱

۱-۳-۳ نگارشهای دیگر مسأله ۱۲

۱-٤ کوتاهترین مسیر در گرافهای تصادفی ۱۳

۱-٤-۱ الگوریتمهای مبتنی بر اتوماتای یادگیر برای حل کوتاهترین مسیر در گرافهای تصادفی ۱٤

۱-٤-۱-۱ کوتاهترین مسیر بین دو راس ۱٤

۱-٤-۱-۲ کوتاهترین مسیر از یک راس به سایر راسها ۱٥

۱-٤-۱-۲-۱ الگوریتم پیشنهادی میسرا و اومن ۱۷

۱-٤-۱-۲-۱ الگوریتم پیشنهادی اصغر قربانی ۱۸

۱-٤-۱-۳ کوتاهترین مسیر بین هر دو راس دلخواه ۲۱

۱-٤-۲ نگارشهای دیگر مسأله ۲۲

۱-٥ درخت اشتاینر مینیمم در گرافهای تصادفی ۲۲

۱-٥-۱ معرفی مسأله درخت اشتاینر مینیمم ۲۲

۱-٥-۲ درخت اشتاینر مینیمم در گرافهای تصادفی ۲٤

۱-٥-۳ نگارشهای دیگر مسأله ۲٥

۱-٦ روشهای نمونه‌گیری مونت-کارلو برای شبیه سازی محیط تصادفی ٢٦
۱-٦-۱ روش تبدیل معکوس ٢٦
۱-٦-۲ روش رد کردن ٢٧
۱-٦-۳ روش نمونه گیری اهمیت ٢٩
۱-٦-٤ زنجیر مارکف مونت کارلو ٣١
۱-٦-٤-۱ نمونه گیری Metropolis-Hasting ٣٤
۱-٦-٤-۲ نمونه گیری Gibbs ٣٦
۱-۷ روش نمونه‌گیری استاندارد ٣٧
۱-۷-۱ بررسی خطای نمونه گیری ٣٨
۱-۷-۱-۱ چند قضیه مهم ٣٩
۱-۷-۱-۲ تخمین امید ریاضی یک متغیر تصادفی ٤٠
۱-۷-۱-۳ تخمین واریانسها ٤١
۱-۷-۲ نمونه‌گیری استاندارد ٤٣
۱-۸ اتوماتای یادگیر ٤٤
۱-۸-۱ تاریخچه اتوماتای یادگیر ٤٦
۱-۸-۲ اتوماتای یادگیر ٤٧
۱-۸-۲-۱ اتوماتای تصادفی ٤٨
۱-۸-۲-۲ محیط ٤٩
۱-۸-۳ معیارهای رفتار اتوماتای یادگیر ٥٠
۱-۸-٤ الگوریتمهای یادگیری ٥٢
۱-۸-٤-۱ الگوریتمهای یادگیری استاندارد ٥٢
۱-۸-٤-۲ تفاوت الگوریتمهای یادگیری با ساختار ثابت و ساختار متغیر ٥٥
۱-۸-٥ اتوماتای یادگیر با عملهای متغیر ٥٥
۱-۸-٦ اتوماتای یادگیر توزیع شده ٥٦
۱-۸-۷ بازی بین اتوماتاهای یادگیر ٥٧
۱-۹ اهداف پایان نامه و ساختار آن (جمع بندی) ٥٨

۲- حل مسأله درخت پوشای مینیمم در گرافهای تصادفی با استفاده از اتوماتاهای یادگیر ٦٢

۲-۱ معرفی ٦٢

۲-۲ تعریف مسأله ٦٢

۲-۳ سیر تلاشها برای حل مسأله درخت پوشای مینیمم در گرافهای تصادفی با استفاده از اتوماتای یادگیر ٦٣

۲-۳-۱ معرفی مفهومی به نام اتوماتای یادگیر اصلاح شده برای حل مسأله ٦٣

۲-۳-۱-۱ اتوماتای یادگیر اصلاح شده ٦٣

۲-۳-۱-۲ الگوریتم پیشنهادی ٦٥

۲-۳-۱-۳ نتایج شبیه سازی ٧٠

۲-۳-۲ ترکیب اتوماتای یادگیر اصلاح شده و الگوریتمهای تولید تصادفی درختهای پوشا ٧٤

۲-۳-۲-۱ الگوریتم پیشنهادی ٧٤

۲-۳-۲-۱-۱ نسخه اول الگوریتم (مقایسه بین وزن بهترین درخت پیدا شده و درخت فعلی) ٧٥

۲-۳-۲-۱-۱ نسخه دوم الگوریتم (مقایسه بین میانگین وزن درختهای پیدا شده و وزن درخت فعلی) ٧٦

۲-۳-۲-۲ بحث در مورد الگوریتم ٧٧

۲-۳-۳ استفاده از اشتراک گذاری اقدامها بین مجموعه‌ای از اتوماتاهای یادگیر برای حل مسأله ٧٨

۲-۳-۳-۱ الگوریتم پیشنهادی ٧٩

۲-۳-٤ تخصیص یک اتوماتا به هر یال برای حل مسأله ٨٢

۲-۳-٤-۱ الگوریتم پیشنهادی اول ٨٢

۲-۳-٤-۲ الگوریتم پیشنهادی دوم ٨٥

۲-۳-٥ استفاده از الگوریتمهای دینامیک موجود برای درختهای پوشای مینیمم برای حل مسأله ٨٥

۲-۳-٥-۱ الگوریتم پیشنهادی ٨٧

۲-۳-٥-۲ بحث در مورد الگوریتم پبشنهادی ٩٢

۲-۳-٦ استفاده از قضایای آماری برای کمک به اتوماتاهای یادگیر برای تشخیص بهتر ٩٣

۲-۳-٦-۱ الگوریتم پیشنهادی نهایی ٩٣

۲-۳-٦-۲ بحث در مورد الگوریتم پیشنهادی نهایی ٩٥

۲-۳-٦-۳ نتایج آزمایشی ٩٨

۲-٤ نتیجه گیری ١٠٣

۳- حل مسأله تطابق وزندار ماکزیمم در گرافهای تصادفی با استفاده از اتوماتاهای یادگیر..... ۱۰۵

۳-۱ معرفی..... ۱۰۵

۳-۲ تعریف مسأله ۱۰۵

۳-۳ الگوریتم پیشنهادی ۱۰۶

۳-٤ بحث در مورد الگوریتم پیشنهادی..... ۱۱۱

۳-۵ نتایج آزمایشی ۱۱۳

۳-٦ نتیجه گیری و کارهای آینده..... ۱۱٤

٤- حل مسأله کوتاهترین مسیر تک مبداء در گرافهای تصادفی با استفاده از اتوماتاهای یادگیر..... ۱۱۷

٤-۱ معرفی..... ۱۱۷

٤-۲ تعریف مسأله ۱۱۷

٤-۳ الگوریتم پیشنهادی ۱۱۸

٤-٤ بحث در مورد الگوریتم پیشنهادی..... ۱۲۲

٤-۵ نتایج آزمایشی ۱۲٦

٤-٦ بحث در مورد مسأله پستچی چینی در گرافهای تصادفی..... ۱۳٦

٤-٦-۱ معرفی مسأله پستچی ۱۳٦

٤-٦-۲ خصوصیات مسأله پستچی..... ۱۳۸

٤-٦-۳ مشکلات پیشرو در حل مسأله پستچی در گرافهای تصادفی..... ۱۳۹

٤-٦-٤ چگونه می‌توانیم مسأله پستچی را در گرافهای تصادفی حل کنیم..... ۱٤۰

٤-۷ نتیجه گیری..... ۱٤۱

۵- حل مسأله درخت اشتاینر مینیمم در گرافهای تصادفی با استفاده از اتوماتاهای یادگیر..... ۱٤٤

۵-۱ معرفی..... ۱٤٤

۵-۲ تعریف مسأله ۱٤۵

۵-۳ الگوریتم پیشنهادی ۱٤۵

۵-٤ بحث در مورد الگوریتم پیشنهادی..... ۱٤۹

۵-۵ نتایج آزمایشی ۱۵۰

۵-٦ نتیجه گیری و کارهای آینده ۱٦۰
٦- نتیجه گیری و کارهای آینده ۱٦۳
مراجع ۱٦٦
واژه نامه ۱۷٦
انگلیسی به فارسی ۱۷٦
فارسی به انگلیسی ۱۸۱
پیوست الف- حل تقریبی مسأله دسته ماکزیمال در گرافها با استفاده از اتوماتای یادگیر توزیع شده ۱۸۸
۱- معرفی ۱۸۸
۲- الگوریتم یادگیری پیشنهادی برای اتوماتای یادگیر ۱۹۰
۲-۱ الگوریتم یادگیری خطی ۱۹۰
۲-۲ الگوریتم یادگیری دنباله رو ۱۹۱
۲-۳ الگوریتم یادگیری پیشنهادی ۱۹۲
۳- الگوریتم پیشنهادی برای حل مسأله ۱۹۲
٤- نتایج آزمایشی ۱۹۷
۵- نتیجه گیری ۱۹۹
پیوست ب- ساختار و تابع توزیع یالهای بعضی از گرافهای تصادفی ۲۰۰
پیوست پ- کدهای سورس برنامه‌ها ۲۰۳
کد سورس درخت پوشای مینیمم ۲۰۳
کد سورس تطابق وزندار ماکزیمم ۲۱۱
کد سورس کوتاهترین مسیر بین یک مبدا و بقیه راسها ۲۱۹
کد سورس درخت اشتاینر مینیمم ۲۲۸

فهرست شکلها

شکل (۱-۱) شرح چگونگی رد /قبول یک نمونه با استفاده از یک عدد تصادفی ۲۸

شکل (۱-۲) رابطه‌ی بین اتوماتا و محیط احتمالی ۵۰

شکل (۱-۳) اتوماتای یادگیر توزیع شده (DLA) با ۳ اتوماتای یادگیر ۵۷

شکل (۱-٤) ارتباط بین اتوماتاهای یادگیر و محیط یک در بازی بین اتوماتاهای یادگیر ۵۸

شکل (۲-۱) ارتباط بین اتوماتای یادگیر اصلاح شده و محیط ٦٤

شکل (۲-۲) الگوریتم پیشنهادی برای حل درخت پوشای مینیمم با استفاده از اتوماتای یادگیر اصلاح شده ٦۹

شکل (۲-۳) ترکیب اتوماتای یادگیر اصلاح شده و الگوریتمهای تولید تصادفی درختهای فراگیر برای حل مسأله درخت فراگیر مینیمم (مقایسه بین وزن بهترین درخت پیدا شده و درخت فعلی) ۷٦

شکل (۲-٤) ترکیب اتوماتای یادگیر اصلاح شده و الگوریتمهای تولید تصادفی درختهای فراگیر برای حل مسأله درخت فراگیر مینیمم (مقایسه بین میانگین وزن درختهای پیدا شده و وزن درخت فعلی) ۷۷

شکل (۲-۵) الگوریتم استفاده از اشتراک گذاری اقدامها بین مجموعه‌ای از اتوماتاهای یادگیر برای حل مسأله درخت پوشای مینیمم ۸۱

شکل (۲-٦) الگوریتم تخصیص یک اتوماتای یادگیر به هر یال برای حل مسأله درخت پوشای مینیمم (الگوریتم پیشنهادی اول) ۸٤

شکل (۲-۷) الگوریتم تخصیص یک اتوماتای یادگیر به هر یال برای حل مسأله درخت پوشای مینیمم (الگوریتم پیشنهادی دوم) ۸٦

شکل (۲-۸) درخت پوشای مینیمم یک گراف تصادفی ۸۷

شکل (۲-۹) تغییرات درخت در حالت ج ۹۰

شکل (۲-۱۰) تغییرات درخت در حالت د ۹۱

شکل (۲-۱۱) استفاده از الگوریتمهای دینامیک درختهای پوشای مینیمم برای حل مسأله درخت پوشای مینیمم در گرافهای تصادفی ۹۲

شکل (۲-۱۲) الگوریتم پیشنهادی برای حل مسأله درخت پوشای مینیمم در گرافهای تصادفی ۹۷

شکل (۲-۱۳) الف- درخت پوشای مینیمم یک درخت تصادفی ب- اتوماتاهای یادگیر بعد از اجرای الگوریتم به اقدام مورد نظری که باید همگرا شوند (به راس ریشه هیچ اتوماتای یادگیری تخصیص نداده شده است).....۹۸

شکل (۲-۱٤) گراف تصادفی G در آزمایش مربوط به درخت پوشای مینیمم.....۹۸

شکل (۲-۱۵) مقایسه بین تعداد نمونه‌های گرفته شده توسط الگویتم پیشنهادی و روش نمونه گیری استاندارد ... ۱۰۲

شکل (۳-۱) الگوریتم پیشنهادی برای حل مسأله تطابق وزندار ماکزیمم در گرافهای تصادفی..... ۱۱۱

شکل (۳-۲) مقایسه بین تعداد نمونه‌های گرفته شده توسط الگویتم پیشنهادی و روش نمونه گیری استاندارد..... ۱۱۵

شکل (٤-۱) الگوریتم پیشنهادی برای حل مسأله کوتاهترین مسیر تک مبدا در گرافهای تصادفی..... ۱۲۱

شکل (٤-۲) یک گراف نمونه با ٦ راس، مقدار روی یک یال نشان دهنده مقدار مورد انتظار وزن آن یال می‌باشد. ۱۲۳

شکل (٤-۳) (a) شبکه ای از اتوماتاهای یادگیر متناظر با گراف شکل (٤-۲)، یالهای ورودی هر راس اقدامهای ممکن اتوماتای یادگیر آن راس می‌باشد. (b) در این شکل بردار احتمال هر یک از اتوماتاهای یادگیر پس از اجرای الگوریتم نشان داده شده است. (c) اتوماتاهای یادگیر A_4(=LA_4) و اقدام‌های آن است. ۱۲۳

شکل (٤-٤) ساختار گراف تصادفی در آزمایش مربوط به کوتاهترین مسیر تک مبدا ۱۲٦

شکل (٤-۵) درخت کوتاهترین مسیر برای گراف تصادفی شکل (٤-٤)..... ۱۲٦

شکل (٤-٦) تعداد کل نمونه‌گیری از یالهای گراف تصادفی شکل (٤-٤) برای دو نسخه الگوریتم پیشنهادی به ازای مقادیر مختلف پارامتر پاداش بین ۰,۰۳ و۰,۵. ..... ۱۳۰

شکل (٤-۷) درصد همگرایی الگوریتم به درخت کوتاهترین مسیر در گراف تصادفی شکل (٤-٤)برای دو نسخه الگوریتم پیشنهادی به ازای مقادیر مختلف پارامتر پاداش بین ۰,۰۳ و۰,۵. ..... ۱۳۰

شکل (٤-۸) درصد همگرایی الگوریتم به درخت کوتاهترین مسیر برای گراف تصادفی شکل (٤-٤) در دو نسخه الگوریتم پیشنهادی به ازای مقادیر مختلف پارامتر پاداش بین ۰,۰۰۲ و ۰,۰۳ . ..... ۱۳۱

شکل (٤-۹) تعداد کل نمونه‌گیری از یالهای گراف تصادفی شکل (٤-٤) برای دو نسخه الگوریتم پیشنهادی به ازای مقادیر مختلف پارامتر پاداش بین ۰,۰۰۲ و ۰,۰۳..... ۱۳۱

شکل (٤-۱۰) مقایسه بین تعداد نمونه‌گیری از یالهای گراف تصادفی به ازی مقادیر مختلف پارامتر یادگیر بین ۰,۰۰۲ و ۰,۰۳ در الگوریتم پیشنهادی و الگوریتم قربانی. (v.1 یعنی نسخه اول و v.2 یعنی نسخه دوم)... ۱۳۵

شکل (٤-١١) مقایسه بین تعداد نمونه‌گیری از یالهای گراف تصادفی به ازی مقادیر مختلف پارامتر یادگیر بین ۰,۰۳ و ۰,۵ در الگوریتم پیشنهادی و الگوریتم قربانی. ۱۳۵

شکل (٤-١٢) مقایسه بین درصد همگرایی الگوریتم پیشنهادی و الگوریتم قربانی به درخت کوتاهترین مسیر به ازای مقادیر مختلف پارامتر یادگیری بین ۰,۰۰۲ و ۰,۲..... ۱۳٦

شکل (٤-١٣) الگوریتم Edmonds و Johnson برای حل مسأله پستچی چینی (کلیات) ۱۳۹

شکل (٥-١) الگوریتم پیشنهادی برای حل مسأله درخت اشتاینر مینیمم در گرافهای تصادفی ۱٤۹

شکل(٥-٢) ساختار گراف تصادفی در آزمایش مربوط به درخت اشتاینر مینیمم (گراف Alex2-B) ۱٥۱

شکل (٥-٣) Alex2-B با مجموعه نقاط ترمینال T={1, 3, 4, 6, 9} و ریشه فرضی 6..... ۱٥۳

شکل (٥-٤) Alex2-B با مجموعه نقاط ترمینال T={1, 5, 8, 9} و ریشه فرضی 1 ..... ۱٥۳

شکل (٥-٥) Alex2-B با مجموعه نقاط ترمینال T={2, 3, 4, 7, 9} و ریشه فرضی 4..... ۱٥۳

شکل (٥-٦) Alex2-B با مجموعه نقاط ترمینال T={5, 7, 9} و ریشه فرضی 7..... ۱٥۳

شکل (٥-٧) نمودارهای مربوط به خطای نسبی متوسط، نسبت به تعداد نمونه گیریهای انجام شده برای چهار مجموعه ترمینالی مختلف در نظر گرفته شده در شکلهای (٥-٣) تا (٥-٦) ۱٥۷

شکل (٥-٨) تغییرات خطای نسبی در مقابل پارامتر یادگیری برای مجموعه‌های ترمینالی در نظر گرفته شده در شکلهای (٥-٣) تا (٥-٦) ۱٥۸

شکل (٥-٩) نمودار نرخ همگرایی الگوریتم در مقابل تغییرات پارمتر یادگیری برای مجموعه‌های ترمینالی در نظر گرفته شده در شکلهای (٥-٣) تا (٥-٦) ۱٥۸

شکل (٥-١٠) نمودار مربوط به مقایسه تعداد نمونه‌های گرفته شده توسط الگوریتم پیشنهادی و روش نمونه گیری استاندارد برای چهار مجموعه ترمینالی مختلف ۱٥۹

شکل (پیوست الف-١) شبه کد الگوریتم پیشنهادی برای حل مسأله دسته ماکزیمال..... ۱۹٦

شکل (پیوست ب-١) گراف Alex1..... ۲۰۰

شکل (پیوست ب-٢) گراف Alex2..... ۲۰۱

شکل (پیوست ب-٣) گراف Alex3..... ۲۰۲

فهرست جدولها

جدول (۱-۱) الگوریتم نمونه گیری تبدیل معکوس ۲۷

جدول (۱-۲) الگوریتم نمونه گیری رد کردن ۲۹

جدول (۱-۳) الگوریتم نمونه گیری اهمیت ۳۱

جدول (۱-٤) الگوریتم نمونه گیری Metropolis-Hasting ۳۵

جدول (۱-۵) الگوریتم نمونه گیری Gibbs ۳۷

جدول (۱-٦) یک فاصله اطمینان $100(1-\alpha)\%$ برای پارامترهای امید ریاضی و واریانس ٤۳

جدول (۲-۱) متوسط تعداد نمونه‌های گرفته شده و درصد همگرایی در الگوریتم مبتنی بر اتوماتای یادگیر اصلاح شده برای گرافهای Alex1-A و Alex1-B ۷۰

جدول (۲-۲) متوسط تعداد نمونه‌های گرفته شده و درصد همگرایی در الگوریتم مبتنی بر اتوماتای یادگیر اصلاح شده برای گرافهای برای گرافهای Alex2-A و Alex2-B ۷۱

جدول (۲-۳) متوسط تعداد نمونه‌های گرفته شده و درصد همگرایی در الگوریتم مبتنی بر اتوماتای یادگیر اصلاح شده برای گرافهای Alex3-A و Alex3-B ۷۱

جدول (۲-٤) تعداد نمونه‌های مورد نیاز در روش نمونه‌گیری استاندارد برای گراف Alex2-B ۷۲

جدول (۲-۵) تعداد نمونه‌های مورد نیاز در روش نمونه‌گیری استاندارد برای گراف Alex3-B ۷۳

جدول (۲-٦) تابع توزیع احتمال وزن یالهای گراف شکل (۲-۱٤) ۹۹

جدول (۲-۷) متوسط تعداد نمونه‌های گرفته شده از یالهای گراف شکل (۲-۱٤) در روش نمونه گیری استاندارد ۱۰۰

جدول (۲-۸) متوسط تعداد کل نمونه‌گیری در نمونه گیری استاندارد از گراف تصادفی شکل (۲-۱٤) ۱۰۰

جدول (۲-۹) متوسط تعداد نمونه‌های گرفته شده درصد همگرایی در الگوریتم پیشنهادی ۱۰۲

جدول (۳-۱) متوسط تعداد نمونه‌های گرفته شده و درصد همگرایی در الگوریتم پیشنهادی ۱۱۵

جدول (٤-۱) تابع توزیع وزن یالهای گراف تصادفی شکل (٤-٤) ۱۲۷

جدول (٤-۲) مقدار مورد انتظار وزن یالهای گراف شکل (٤-٤) ۱۲۷

جدول (٤-٣) امید ریاضی فاصله راسها از راس مبدا در گراف تصادفی شکل (٤-٤) ١٢٧
جدول (٤-٤) متوسط تعداد نمونه‌های گرفته شده و درصد همگرایی در الگوریتم پیشنهادی (نسخه اول) ١٢٨
جدول (٤-٥) متوسط تعداد نمونه‌های گرفته شده و درصد همگرایی در الگوریتم پیشنهادی (نسخه دوم) ١٢٩
جدول (٤-٦) متوسط تعداد نمونه‌های گرفته شده و درصد همگرایی در الگوریتم قربانی (نسخه اول الگوریتم) ... ١٣٣
جدول (٤-٧) متوسط تعداد نمونه‌های گرفته شده و درصد همگرایی در الگوریتم قربانی (نسخه دوم الگوریتم)... ١٣٤
جدول (٥-١) تعداد نمونه‌های مورد نیاز در روش نمونه‌گیری استاندارد برای گراف Alex2-B ١٥٢
جدول (٥-٢) اطلاعات به دست آمده از الگوریتم پیشنهادی برای شکل (٥-٣) ١٥٤
جدول (٥-٣) اطلاعات به دست آمده از الگوریتم پیشنهادی برای شکل (٥-٤) ١٥٥
جدول (٥-٤) اطلاعات به دست آمده از الگوریتم پیشنهادی برای شکل (٥-٥) ١٥٥
جدول (٥-٥) اطلاعات به دست آمده از الگوریتم پیشنهادی برای شکل (٥-٦) ١٥٦
جدول(پیوست الف-١) نتایج اجرای الگوریتم پیشنهادی روی تعدادی از گرافها ١٩٨
جدول(پیوست الف-٢) مقایسه الگوریتم پیشنهادی با الگوریتم ارائه شده در [Ali06] ١٩٨
جدول(پیوست الف-٣) مقایسه الگوریتم پیشنهادی با بعضی الگوریتمها ١٩٨
جدول (پیوست ب-١) تابع توزیع یالهای گراف Alex1-B ٢٠٠
جدول (پیوست ب-٢) تابع توزیع یالهای گراف Alex1-A ٢٠٠
جدول (پیوست ب-٣) تابع توزیع یالهای گراف Alex2-B ٢٠١
جدول (پیوست ب-٤) تابع توزیع یالهای گراف Alex2-A ٢٠١
جدول (پیوست ب-٥) تابع توزیع یالهای گراف Alex3-B ٢٠٢
جدول (پیوست ب-٦) تابع توزیع یالهای گراف Alex3-A ٢٠٢

فصل اول

مقدمه

۱- مقدمه

در این فصل ابتدا گرافهای تصادفی را به صورت رسمی در بخش اول معرفی می‌کنیم که کلیه کارهای انجام گرفته در این پایان نامه روی این نوع از گرافهاست که مدل مناسبی برای شبکه‌های واقعی که وزن لینکها به طور مداوم و پیوسته تغییر می‌کند می‌باشد. در چهار بخش بعدی به ترتیب چهار مسأله درخت پوشای مینیمم، تطابق وزندار ماکزیمم، کوتاهترین مسیر بین یک راس و سایر رسها و درخت اشتاینر را در گرافهای تصادفی به اختصار مورد بررسی قرار می‌دهیم و کلیاتی در مورد کارهای مرتبط قبلی در صورت وجود و کارهایی که ما در این پایان نامه انجام داده‌ایم را ذکر می‌کنیم.

در بخش ششم روشهای نمونه‌گیری مونت-کارلو را که برای شبیه سازی محیط تصادفی به کار می‌روند را به اختصار توضیح خواهیم داد و چند روش ساده و متداول این کار را بیان خواهیم کرد.

در بخش هفتم روش نمونه‌گیری استاندارد را که برای ارزیابی الگوریتمهای پیشنهادی که قبلا الگوریتم دیگری برای آنها ارائه نشده را بیان خواهیم کرد.

در بخش هشتم این فصل اتوماتای یادگیر، اتوماتای یادگیر توزیع شده و بازی بین اتوماتاهای یادگیر را شرح خواهیم داد.

بخش آخر اهداف پایان نامه و ساختار آن را بیان می‌کند و یک جمعبندی از مطالب بخشهای قبلی را شامل می‌شود.

۱-۱ گراف تصادفی

یک گراف تصادفی[1] G را توسط سه‌تایی $G =< V, E, F >$ تعریف می‌کنیم که در آن $V = \{1,2,3,...,n\}$ مجموعه راسهاست و $E \subset V \times V$ مجموعه یالها را معین می‌کند و ماتریس F که یک ماتریس $n \times n$ است توزیع احتمالی وزن یالهای گراف را مشخص می‌کند. وزن (طول) یال (i, j) را با متغیر تصادفی C_{ij} که یک متغیر تصادفی مثبت فرض می‌شود نشان می‌دهیم و f_{ij} تابع توزیع (چگالی) احتمال C_{ij} در نظر گرفته می‌شود. در صورتی که گراف تصادفی را غیر جهتدار بنامیم آنگاه یالها غیرجهت دار بوده و داریم $C_{ij} = C_{ji}$ برای هر $(i, j) \in E$.

[1] Stochastic graph

در کلیه کارهایی که در این پایان نامه انجام داده‌ایم سه فرض را زیر را برای گرافهای تصادفی در نظر گرفته‌ایم:

۱. وزن یالها از یکدیگر مستقل باشد.
۲. تابع توزیع احتمال وزن یالها ناشناخته است و فقط نمونه‌های این توزیعها در دسترس هستند.
۳. تابع توزیع احتمال یالها هرچند که برای ما شناخته شده نیست ولی در طول زمان تغییر نمی‌کند.

قرارداد: هر جا که از اصطلاح گراف تصادفی استفاده می‌کنیم منظور گرافی با مشخصات بالا می‌باشد مگر آن که خلاف آن ذکر شود.

۱-۱-۱ چه الگوریتمهایی برای گرافهای تصادفی مناسبند؟

باید توجه کرد از آنجا که وزن یالهای گراف تصادفی به طور همزمان و پیوسته تغییر می‌کنند الگوریتمهای موجود در گرافهای قطعی یعنی گرافهایی که وزن یالها در آنها دقیقا مشخص است نمی‌توانند جوابگو باشند زیرا که در هر لحظه جواب بهینه در گراف تغییر می‌کند. در چنین شرایطی باید به دنبال جوابی بگردیم که به طور متوسط در طول زمان بهتر از بقیه جوابها باشد. به عبارت دیگر باید به دنبال جوابی باشیم که مقدار مورد انتظار آنها (امید ریاضی آنها) نسبت به بقیه جوابها بهینه تر باشد.

الگوریتمهای دینامیک موجود در گرافها که بعد از هر تغییر در وزن یکی از یالهای گراف سعی مجدد در یافتن جواب بهینه با استفاده از اطلاعات قبلی دارند نیز به همان دلیل که وزن یالها به طور همزمان و پیوسته تغییر می‌کنند نمی‌توانند جوابگو باشند.

از آنجا که در گرافهای تصادفی تابع توزیع وزن یالها مشخص نیست باید الگوریتم از یک ابزار یادگیری استفاده کند که در تعامل با محیط ناشناخته بتواند بعضی از پارامترهای آن را یاد بگیرد. این ابزار می‌بایست جزو ابزار یادگیری تقویتی باشد که در طول زمان بتواند جواب بهینه را با احتمالی تشخیص دهد. ابزار یادگیری تقویتی که ما از آن استفاده می‌کنیم اتوماتای یادگیر می‌باشد. هر چند که‌این ابزار دارای ساختار بسیار ساده‌ای می‌باشد ولی از طریق همکاری بین شبکه‌ای از اتوماتای یادگیر می‌تواند مسائل پیچیده‌ای را حل کند.

نکته مهمی که در مورد الگوریتمهای گرافهای تصادفی وجود دارد این است که الگوریتم در صورتی که بتواند به یک جواب بهینه همگرا شود دیگر تغییرات وزن یالها تاثیری در جواب بهینه ندارد و جواب بهینه همچنان محفوظ خواهد بود و دیگر با تغییر وزن یالها نیاز به هیچ هزینه زمانی برای به روزرسانی جواب نخواهیم داشت.

۱-۲ درخت پوشای مینیمم در گرافهای تصادفی

۱-۲-۱ معرفی مسأله درخت پوشای مینیمم

درخت فراگیر مینیمم یک گراف وزندار ، درخت فراگیری است که بین همه درختهای فراگیر آن گراف دارای کمترین وزن باشد. وزن تخصیص داده شده به یک یال می‌تواند هزینه یال، زمان پیمایش یال، طول و یا غیره بنا به محتوای مورد نظر باشد. درختهای فراگیر مینیمم دارای کاربردهای فراوانی در طراحی سیستمهای فیزیکی می‌باشند. همچنین در کاربردهایی نظیر آنالیز خوشه‌بندی آماری[1]، شناسایی گفتار و پردازش تصاویر مورد استفاده هستند[Hut06].

در حالتی که وزن یالها قطعی باشد الگوریتمهای حریصانه ای چون کروسکال(۱۹۵٦)، پریم-دایجکسترا (پریم ۱۹۵۷، دایجکسترا ۱۹۵۹) و سولین(بروفکا) (سولین۱۹۷۷) ارائه شده است که در اصل مشهورترین الگوریتمهای ارائه شده برای حل مسأله درخت فراگیر مینیمم یعنی یافتن درخت فراگیری چون T از G با کمترین مجموع وزن، می‌باشند [Ahr02]. نسخه موازی شده این الگوریتمها نیز ارائه شده است: پیچیدگی نسخه موازی شده الگوریتم کروسکال با $p = \lceil \log\ m \rceil$ پردازنده برابر با $O(m)$ می‌باشد. پیچیدگی الگوریتم پریم با p پردازنده که به صورت hypercube آرایش شده اند برابر است با $\theta(n^2/p)+\theta(n\log p)$ وپیچیدگی نسخه موازی شده الگوریتم سولین با p پردازنده برابر است با $O(\log n(n^2/p+n/p+n+p))$ [Ahr02].

همچنین الگوریتمهای توزیع شده برای این مسأله ارائه گردیده است که بعضی از آنها به شکل دقیق درخت پوشای مینیمم را محاسبه می‌کنند [Gal83][Chn85][Awr87][Gar98] و بعض به شکل تقریبی ولی در زمان بسیار کمتر درخت پوشای مینیمم را به دست می‌دهند[Khn06].

[1] Statistical cluster analysis