دانشگاه صنعتی شریف
دانشکده‌ی مهندسی کامپیوتر

پایان‌نامه‌ی کارشناسی ارشد
مهندسی نرم‌افزار

عنوان:

قالب استاندارد برای نگارش پایان‌نامه‌ها

نگارش:

حمید ضرابی‌زاده

استاد راهنما:

استاد راهنمای پروژه

شهریور ۱۳۹۹


بسم الله الرحمن الرحیم

چکیده

امروزه استفاده از شبکه کانال های پرداخت[۱] مبتنی بر بلاکچین به عنوان یکی از عملی ترین راه حل های مشکل عدم مقیاس پذیری بلاکچین بسیار مورد توجه قرار گرفته است. کاربران با استفاده از شبکه کانال های پرداخت برای تراکنش های روزمره،در عین اینکه از تمام تضمین های امنیتی و محرمانگی بلاکچین بهره مند میشوند، میتوانند از پرداخت کارمزد های سرسام آور بلاکچین خودداری کنند.

نحوه کار کانال پرداخت به صورت خلاصه به شرح زیر است. دو نفر برای ایجاد یک کانال پرداخت باید با فرستادن تراکنش مخصوصی به بلاکچین، مقداری سپرده برای کانال خود ذخیره کنند. بعد از ایجاد کانال دو نفر میتوانند تا سقف سپرده خود تراکنش برون بلاکچینی [۲] برای هم بفرستند و چون این تراکنش ها به صورت محلی[۳] و بدون مراجعه به بلاکچین انجام میشوند بسیار سریع هستند و کارمزد آن ها ناچیز است. در انتها وقتی طرفین تصمیم به بستن کانال خود میگیرند با ارسال تراکنش دیگری به بلاکچین میتوانند سپرده خود را آزاد کنند. بدین ترتیب با تنها دو تراکنش درون بلاکچینی[۴] یکی برای ایجاد کانال و دیگری برای بستن کانال، امکان ارسال صدها تراکنش برون بلاکچینی فراهم میشود.

یکی از مهم ترین محدودیت های کانال پرداخت این است که افراد امکان اضافه کردن سپرده به کانال را فقط و فقط در هنگام ایجاد کانال دارند و اگر بعدا تصمیم به افزایش سپرده خود بگیرند باید کانال را بسته و کانال جدیدی ایجاد کنند که امری هزینه بر است. بنابراین کاربران تمایل دارند مقدار سپرده کافی در کانال از همان ابتدا قرار دهند اما از طرف دیگر نباید بیش از اندازه هم در کانال پول بگذارند زیرا امکان استفاده از این پول را تا بستن کانال نخواهند داشت. در نتیجه کاربران هنگام ایجاد کانال با یک مسأله تصمیم گیری آنلاین روبرو هستند. اما در عمل مساله از این هم پیچیده تر است زیرا میلیون ها کاربر با کانال پرداخت های دو به دویی که تشکیل میدهند، شبکه عظیمی از کانال های پرداخت [۵] را تشکیل میدهند. در این شبکه هر دو نودی که که یک کانال پرداخت مشترک دارند میتوانند بی واسطه برای هم تراکنش بفرستند اما نود هایی که کانال مستقیم با هم ندارند باید با استفاده از سایر نود های شبکه به عنوان واسطه، تراکنش خود را در شبکه مسیریابی و ارسال کنند. به طور مثال شبکه ای به صورت A-B-C را در نظر بگیرید که در آن نود های A و C کانال پرداخت مشترک ندارند اما هر دو با

[۱] payment channel
[۲] بدون مراجعه به بلاکچین off-chain transaction
[۳] local
[۴] on-chain
[۵] payment channel networks

B کانال مشترک دارند؛ در این شبکه A میتواند برای B پول بفرستد و B همان پول را به C ارسال کند و تراکنش A به C با یک واسطه انجام خواهد شد. بنابراین نود های موجود در شبکه میتوانند در دو نقش کاربر(فرستنده یا گیرنده) یا سرویس دهنده(واسطه) ایفای نقش کنند. نود های واسطه در ازای انتقال تراکنش های کاربران کارمزد دریافت میکنند پس تمایل دارند که تا حد امکان تراکنش های بیشتری را مسیریابی کنند؛ اما از طرفی اگر واسطه ها حریصانه تمام تراکنش های کاربران را مسیریابی کنند، کانال هایشان خالی از پول میشود. مثلا در مثال بالا فرض کنید A قصد دارد تعداد تراکنش زیادی برای C بفرستد، اگر B تمام این تراکنش ها را مسیریابی کند، هیچ پولی در کانال B-C برای او باقی نخواهد بود و در عوض Bمقدار زیادی پول در کانال A-B مقدار زیادی پول خواهد داشت. در چنین شرایطی میگوییم کانال B نامتعادل شده است. نامتعادل شدن کانال امر مطلوبی نیست زیرا مانع انتقال تراکنش های بعدی در جهت خالی شده از پول میشود. پس نود های واسطه شبکه کانال های پرداخت در هر لحظه با یک تصمیم گیری آنلاین روبرو هستند؛ اینکه کدام یک از تراکنش های کاربران را انتقال دهند. بنابراین در مجموع میبینیم که نود های شبکه کانال های پرداخت چه هنگام ایجاد کانال و چه بعدا د زمان ارسال تراکنش های برون بلاکچینی باید مدام تصمیمات آنلاینی در خصوص مدیریت کانال خود بگیرند.

نود های شبکه کانال های پرداخت نیاز به الگوریتمی برای مدیریت کانال خود دارند. این الگوریتم باید آنی باشد به این معنی که الگوریتم برای اتخاذ تصمیمات زمان زیادی ندارد. طراحی یک الگوریتم بهینه آنلاین که درباره مدیریت سپرده های نود ها تصمیم گیری میکند نه تنها در این حوزه مورد نیاز است بلکه میتواند در حوزه های دیگر همچون شبکه های مخابراتی برای حل مسأله admission control هم سود بخش باشد. در این پایان نامه الگوریتم آنلاینی برای مدیریت یک تک کانال پرداخت طراحی میکنیم. الگوریتم ما یک الگوریتم آنلاین است به این معنی که هیچ فرض خاصی روی توزیع تراکنش های آینده ندارند و تنها با اطلاعات گذشته و لحظه حال تصمیمی اتخاذ میکند. در این پایان کران بالای هزینه الگوریتممان را برای بدترین دنباله تراکنش [۶] ممکن اثات میکنیم؛ و در نهایت با پیاده سازی نشان میدهیم که الگوریتم ما در عمل بسیار بهتر از تضمین تئوری اثبات شده عمل میکند و همچنین دو heuristic با الهام از الگوریتم اصلی طراحی میکنیم که در عمل هزینه را تا نصف هزینه الگوریتم اصلی کاهش میدهد.

کلیدواژه‌ها: بلاکچین، شبکه کانال های پرداخت، الگوریتم آنلاین، admission control

---

[۶] worst-case analysis

فهرست مطالب

فهرست تصاویر

فهرست جداول

فصل ۱

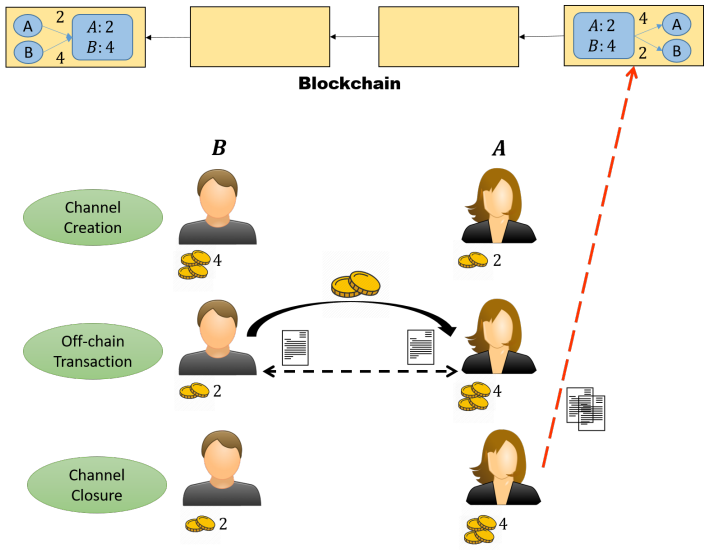
مقدمه

بلاکچین هایی همچون بیتکوین به دلیل ماهیت توزیع شده والگوریتم اجماع پیچیده و وقت گیری که دارند با مشکل عدم مقیاس پذیری روبرو هستند. عدم مقیاس پذیری به این معنی است که سیستم نمیتواند تعداد بسیار زیاد تراکنش را پردازش کند. به طور مثال بلاکچین بیتکوین تنها میتواند ۷ تراکنش در ثانیه را پردازش کند در حالیکه رقبای متمرکز بلاکچین همچون visa بیش از هزاران تراکنش را در هر ثانیه پردازش میکنند. به علاوه، حتی وقتی تراکنش ها وارد بلاکچین میشوند تایید شدن آن ها معمولا حداقل چند دقیقه به طول می انجامد، به طور مثال در بلاکچین بیتکوین نزدیک یک ساعت طول میکشد تا یک تراکنش تأیید نهایی شود. یکی از مورد استقبال ترین راه حل هایی که برای حل مشکل مقیاس ناپذیری و کندی بلاکچین ارائه شده است استفاده از شبکه کانال های پرداخت[۱] است. شبکه کانال های پرداخت اولین بار با پیاده سازی Lightning Network روی بلاکچین بیتکوین معرفی شد. [؟] بعد ها شبکه کانال های پرداخت Raiden هم با الهام از Lightning Network بر بلاکچین اتریوم[۲] توسعه داده شد. [؟] کاربران میتوانند با ارسال یک تراکنش ایجاد کانال [۳] روی بلاکچین، یک کانال پرداخت ایجاد کنند. با این تراکنش در واقع طرفین کانال پرداخت، مقداری پول را در این کانال پرداخت به سپرده میگذارند. پس از ایجاد کانال، طرفین میتوانند بدون مراجعه به بلاکچین و با رد و بدل کردن تعدادی امضای دیجیتال برای هم تراکنش محلی فوری[۴] با کارمزد بسیار اندک و بفرستند. مبادله امضاهای دیجیتال برای حفظ امنیت مالی طرفین الزامی است.

---

[۱] payment channel network
[۲] Ethereum
[۳] channel creation
[۴] instant

شکل ؟؟ نحوه کار یک کانال پرداخت را نشان میدهد. ابتدا کاربر A ۲ واحد پول و کاربر B ۴ واحد پول سپرده میکند و یک کانال پرداخت میسازند. تراکنش ایجاد کانال روی بلاکچین قرار میگیرد. پس از ایجاد کانال امکان ارسال تراکنش برون بلاکچینی فراهم میشود. B میخواهد برای A دو واحد پول واریز کند پس A و B امضاهای دیجیتال رد و بدل میکنند و دو واحد پول به صورت برون بلاکچینی به موجودی A اضافه میشود. پس از مدتی A تصمیم میگیرد کانال را ببندد؛ از امضاهای رد و بدل شده پیشین استفاده میکند تا یک تراکنش بستن کانال ایجاد کند. پس از اجرای این تراکنش هر کدام از A و B سه واحد پول میگیرند. توجه کنید که در مرحله دوم A و B میتوانند به تعداد نامحدود تراکنش برون بلاکچینی ایجاد کنند. پس با دو تراکنش درون بلاکینی، امکان ایجاد تعداد نامحدود ترامنش برون بلاکچینی ارزان و سریع فراهم شد. اما باید توجه کرد که مجموع موجودی A و B که به آن ظرفیت کانال [۵] می گویند همواره عدد ثابت ۶ است و قابل افزایش یا کاهش نیست.



شکل ۱-۱: نحوه شکل گیری، استفاده و بستن یک کانال پرداخت

از اتصال کاربران مختلف با کانال های پرداخت یک شبکه از کانال های پرداخت ایجاد میشود که میتواند افرادی که کانال پرداخت مستقیم به هم ندارند را هم با یک یا تعدادی واسطه به هم متصل کند. مثلا ۳ کاربر A-B-C را در نظر بگیرید که A-B و B-C کانال پرداخت دارند. در این صورت A و C اگرچه کانال پرداخت مشترک ندارند اما میتوانند از B به عنوان واسطه استفاده کنند و برای هم تراکنش برون بلاکچینی ارسال کنند؛ بدین صورت که A مقدار پول مورد نظر را برای B می فرستد و B همان مقدار پول را برای C میفرستد. این دو تراکنش ها atomic هستند که به این معنی است که یا هر دو آنها انجام

[۵] capacity

میشوند و یا هر دو برگشت میخورند. معمولا فرد واسطه یعنی B مقداری کارمزد از A میگیرد اما این کارمزد در برابر کارمزد های تراکنش های درون بلاکچینی بسیار ناچیز است و صرفا نقش ایجاد انگیزه برای واسطه ها را دارد. البته گاهی نود های واسطه ممکن است برخی تراکنش ها را به دلایلی رد کنند. مثلا ممکن است اندازه تراکنش بیشتر از موجودی آن نود واسطه در کانال باشد یا اینکه موجودی نود واسطه را در حد غیر قابل قبولی کاهش دهد و یا اینکه میزان کارمزد آن مطلوب نود واسطه نباشد.

یکی از مشکلات بسیار مهم شبکه کانال های پرداخت این است که بعد از ایجاد کانال هیچ راهی برای افزودن سپرده به کانال وجود ندارد. مثلا در مثال بالا در کانال A-B فرض کنید با شروع از سپرده اولیه ۴ – ۲ ،A ۲ تراکنش هر کدام به ارزش ۱ بیتکوین برای B میفرستد؛ پس از انجام این دو تراکنش موجودی آنها در کانال به ترتیب ۶ – ۰ خواهد بود. پس از این تا زمانی که B تراکنشی برای A نفرستد، A نمیتواند تراکنشی بفرستد زیرا موجودی اش صفر است. به کانال پرداختی که در آن موجودی یک نفر صفر(یا بسیار کم است) کانال نامتعادل [۶] میگوییم. کانال های نامتعادل برای کاربران به خصوص برای نود های واسطه اصلا مطلوب نیستند زیرا امکان ایجاد تراکنش از یک سمت کانال را به کل از بین میبرند. در این پایان نامه گاهی نود های واسطه را سرویس دهنده [۷] می نامیم. نود های سرویس دهنده نود هایی هستند که با هدف درآمد سازی به شبکه کانال های پرداخت ملحق میشوند و با ذخیره کردن مقدار چشم گیری سپرده، تعداد زیادی کانال با کاربران زیادی ایجاد میکنند تا تراکنش های آنها را مسیریابی کنند و در ازای آن کارمزد بگیرند. داشتن کانال های نامتعادل توانایی سرویس دهنده ها را در انتقال تراکنش ها از یک جهت کاهش میدهد و برای کسب و کار آنها مشکل ایجاد میکند. Lightning network دو راه حل را برای حل مشکل کانال های نامتعادل پیشنهاد میدهد:

۱. شارژ کردن درون بلاکچینی: در این روش طرفین کانال نامتعادل آن کانال را میبندند و کانال جدیدی با سپرده بیشتر باز میکنند. این عمل باعث ایجاد دو تراکنش درون بلاکچینی میشود. یک تراکنش برای بستن کانال قدیمی و یک تراکنش برای ایجاد کانال جدید.

۲. متعادل کردن برون بلاکچینی: این روش بدون مراجعه به بلاکچین و صرفا با تعدادی تراکنش برون بلاکچینی توزیع سپرده ها را در کانال نامتعادل تغییر میدهد و به نسبت روش قبل ارزان تر است. در بخش ؟؟ به طور مفصل این روش را توضیح میدهیم.

از آنجاییکه هر دو روش بالا هزینه بر هستند و محدودیت هایی را اعمال میکنند، تصمیم گیری بر

[۶] depleted channel
[۷] service provider

سر اینکه چه زمانی کدام یک از آنها انجام گیرد تصمیم سختی است. همچنین توجه کنید که به عنوان یک کاربر یا یک نود واسطه، معمولا نود اطلاعات دقیقی از تراکنش های آینده ندارد و در نتیجه نود ها باید سیاست تصمیم گیری ای را اتخاذ کنند که بر اساس تاریخچه و بدون فرضی روی تراکنش های آینده، تصمیم گیری میکند.

هدف این پایان نامه این است که سیاست آنلاینی طراحی کند یک تک کانال پرداخت را در کلی ترین حالت ممکن در نظر میگیرد و به سوالات زیر که برای بیشینه کردن سود و کمینه کردن هزینه طرفین کانال مطرح میشود پاسخ میدهد:

۱. چه زمانی ایجاد یک کانال پرداخت نسبت به ارسال درون بلاکچینی تراکنش به مقرون به صرفه است؟

۲. اگر تصمیم به ایجاد کانال پرداخت شد، طرفین چه مقدار سپرده باید در آن قرار دهند؟

۳. اگر طرفین میخواهند نقش واسطه را ایفا کنند چه تراکنش هایی را باید بپذیرند و چه تراکنش هایی را نپذیرند؟

۴. اگر کانال پرداخت نامتعادل شد، طرفین باید کدام یک از راه های مقابله با کانال نامتعادل را اتخاذ کنند و سپرده کانال را چقدر باید تغییر دهند؟

۱-۱ اهمیت موضوع

هدف از طراحی شبکه کانال های پرداخت ایجاد بستری ارزان و سریع برای انجام تراکنش های کوچک و روزانه [۸] است. بهره بری کاربران از شبکه کانال های پرداخت تا حد زیادی به نحوه مدیریت کانال توسط آنها و سرویس دهنده ها بستگی دارد. مدیریت نادرست کانال ها توسط کاربران میتواند منجر به نامتعادل شدن کانال های آن ها شود و معمولا هزینه اصلاح یک کانال نامتعادل بسیار زیاد است. همچنین مدیریت نادرست کانال ها توسط سرویس دهنده هاهم به ضرر خود سرویس دهنده ها و هم به ضرر کاربران است. اگر سرویس دهنده ها نتوانند کانال های خود را درست مدیریت کنند، سود آنها کاهش می یابد و انگیزه ای برای ارائه خدمات نخواهند داشت که با توجه به اهمیت حیاتی سرویس دهنده

[۸] micro payment

ها برای شبکه، این امر بسیار مضر است. با بررسی آخرین داده های موجود از Lightning Network [؟] میتوان دید که در سال ۲۰۲۱ حدود ۶۳۰۰ در شبکه وجود دارد که بیش از ۵۰ درصد آن ها تنها از ۱۰ سرویس دهنده خدمات میگیرند. یعنی اگر ۱۰ سرویس دهنده اصلی Lightning Network عملکرد مناسبی نداشته باشند، نیمی از شبکه مختل خواهد شد! در واقع بدون وجودسرویس دهنده ها، امکان ارسال تراکنش های با واسطه از بین میرود و همه کاربران مجبورند کانال های دو به دو با هم ایجاد کنند.

در نتیجه ارائه الگوریتمی که این مسئله مدیریت کانال را در یک مدل واقع بینانه، با کمترین فروض محدود کننده و به صورت بهینه حل کند، بسیار ارزشمند است.

۱-۲ دست آورد های تحقیق

در این پایان نامه برای حل مسأله مدیریت آنلاین کانال های پرداخت، ابتدا از حل تئوری یک نسخه بسیار ساده شده و غیرواقع گرایانه مسأله شروع میکنیم و سپس در دو گام مدل را پیچیده تر واقع گرایانه تر میکنیم طوری که مسأله نهایی تا حد خوبی بیشتر پیچیدگی های کانال های پرداخت در دنیای واقعی را در بر دارد. این دو زیر مسأله به شرح زیر هستند:

۱. **زیر مسأله ۱ (کانال یکطرفه همیشه پذیرنده[۹])** کانال پرداختی ساده و غیرواقع نگرانه ای با دو کاربر A و B را در نظر بگیرید که در آن همیشه فقط A برای B پول میفرستد، یعنی کانال یکطرفه است. همچنین فرض کنید که باید تمام تراکنش ها حتما انجام شود و کاربران امکان رد کردن تراکنش ها را ندارند(اگر A یک کاربر عادی باشد رد تراکنش به این معنی است که A تراکنشش را خارج از کانال پرداخت و از طرق دیگر انجام میدهد و اگر A یک سرویس دهنده باشد رد کردن تراکنش به این معنی است که A تصمیم میگیرد از کارمزد این تراکنش صرف نظر کند و این تراکنش را مسیریابی نکند). همچنین برای ساده سازی فرض کنید که اگر پول A در کانال پرداخت تمام شد، باید کانال را ببندد و کانال جدید باز کند یا به عبارت دیگر تنها راه متعادل کردن کانال، شارژ کردن درون بلاکچینی است و متعادل کردن برون بلاکچینی برای ساده سازی مجاز نیست. این مدل، اولین و ساده ترین مدلی است که بررسی میکنیم و برای آن الگوریتمی

[۹] Unidirectional stream without rejection

مسئله	نسبت رقابتی	کران پایین
Unidirectional stream without rejection	2	2
Unidirectional stream with rejection	$2+\frac{\sqrt{5}-1}{2}$	$2+\frac{\sqrt{5}-1}{2}$
Bidirectional stream	$7+2\log C$	$\theta(\sqrt{\log C})$

جدول ۱-۱: خلاصه نتایج تئوری این پایان نامه. ستون اول نام (زیر)مسأله، ستون دوم نسبت رقابتی و ستون

آنلاین با نسبت رقابتی [۱۰] برابر ۲ ارائه میدهیم و اثبات میکنیم که این بهترین نسبت رقابتی ای ست که یک الگوریتم آنلاین میتواند به آن دست یابد.

۲. **زیر مسأله ۲ (کانال یکطرفه مجاز به رد تراکنش[۱۱])** در این زیر مسأله همانند مدل قبلی جهت تراکنش ها همیشه یکطرفه است اما این بار دارندگان کانال میتوانند تصمیم بگیرند کدام تراکنش ها را انتقال دهند و کدام ها را رد کنند. مشابه مدل قبل متعادل کردن برون بلاکچینی مجاز نیست. برای این مدل الگوریتم آنلاین با نسبت رقابتی $\frac{\sqrt{۵}-۱}{۲}+۲$ ارائه میدهیم و اثبات میکنیم که این الگوریتم بهینه است.

۳. **مسأله اصلی (کانال دوطرفه [۱۲])** در کلی ترین حالت مسأله تراکنش ها در هر دو جهت وجود دارند و صاحبان کانال نه تنها میتوانند تراکنش ها را به دلخواه بپذیرند یا رد کنند بلکه میتوانند از هر دو روش شارژ کردن درون بلاکچینی و متعادل کردن برون بلاکچینی برای متعادل کردن کانال خود استفاده کنند. برای این مدل الگوریتم آنلاین با نسبت رقابتی $۲\log C+۷$ طراحی میکنیم. (C یک عدد ثابت است که بستگی به ویژگی های گراف شبکه کانال های پرداخت دارد و مثلا در `Lightning Network` حدودا برابر ۴ است). همچنین به عنوان کران پایین نشان میدهیم که هیچ الگوریتم آنلاینی با نسبت رقابتی $o(\sqrt{\log C})$ وجود ندارد.

الگوریتم ها و اثبات های تئوری زیرمسأله ۱ و ۲ به عنوان بلوک های سازنده برای حل مسأله اصلی مورد استفاده قرار میگیرد.

[۱۰] `competitive ratio` معیاری است که هزینه یک الگوریتم آنلاین را با هزینه الگوریتم بهینه آفلاین که از پیش به تمام تراکنش های آینده دسترسی دارد، مقایسه میکند. در قسمت ؟؟ به طور مفصل این معیار و نحوه محاسبه آن را توضیح میدهیم.

[۱۱] Unidirectional stream with rejection

[۱۲] Bidirectional stream

۱-۳ ساختار پایان‌نامه

این پایان‌نامه شامل پنج فصل است. فصل دوم دربرگیرنده‌ی تعاریف اولیه‌ی مرتبط با پایان‌نامه است. در فصل سوم مسئله‌ی دورهای ناهمگن و کارهای مرتبطی که در این زمینه انجام شده به تفصیل بیان می‌گردد. در فصل چهارم نتایج جدیدی که در این پایان‌نامه به دست آمده ارائه می‌گردد. در این فصل، مسئله‌ی درخت‌های ناهمگن در چهار شکل مختلف مورد بررسی قرار می‌گیرد. سپس نگاهی کوتاه به مسئله‌ی مسیرهای ناهمگن خواهیم داشت. در انتها با تغییر تابع هدف، به حل مسئله‌ی کمینه کردن حداکثر اندازه‌ی درخت‌ها می‌پردازیم. فصل پنجم به نتیجه‌گیری و پیش‌نهادهایی برای کارهای آتی خواهد پرداخت.

فصل ۲

مفاهیم اولیه

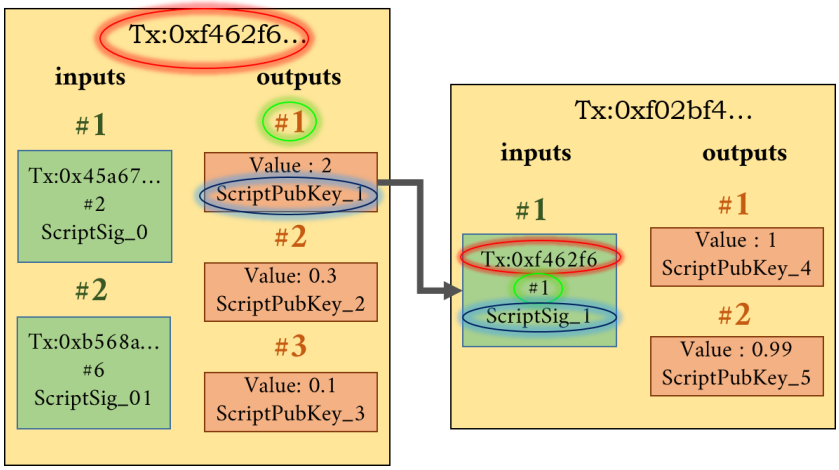
در این فصل مفاهیم اولیه لازم برای فهم مسأله و نتایج پایان نامه را مطرح میکنیم. ابتدا نحوه پردازش تراکنش ها در بلاکچین بیتکوین را توضیح میدهیم و سپس توضیح میدهیم که تراکنش های یک کانال پرداخت چه تفاوتی با تراکنش های عادی درون بلاکچینی دارند و چگونه میتوان تراکنش برون بلاکچینی امن داشت. در این بخش هنگام توضیح جزئیات پروتکل ها بلاکچین بیتکوین و شبکه کانال های پرداخت آن یعنی Lightning network را به عنوان معیار در نظر میگیریم زیرا اولا امروزه Lightning network با داشتن بیش از ۱۷۰۰۰ نود، پرکاربر ترین شبکه کانال های پرداخت موجود است [؟] و ثانیا مفاهیم پایه ای تراکنش های درون بلاکچینی و برون بلاکچینی کمابیش برای تمام بلاکچین ها یکسان است و تنها تفاوت در پروتکل های پیاده سازی شده است، پس تفاوت چندانی ندارد که کدام بلاکچین را به عنوان معیار قرار دهیم.

۲-۱ تراکنش ها در بیتکوین

بیتکوین یک بلاکچین UTXO-based است، در این سیستم هر تراکنش یک یا تعدادی ورودی و یک یا تعدادی خروجی دارد. در هر تراکنش مجموع بیتکوین ورودی ها برابر است با مجموع بیتکوین خروجی ها و کارمزد تراکنش. شکل ؟؟ را ببینید. هر تراکنش یک هش [۱] یکتا دارد که شناسه تراکنش محسوب میشود. هر کدام از ورودی های یک تراکنش یکی از خروجی های یک تراکنش قدیمی تر را خرج میکند.

[۱] hash

هر خروجی دو داده در بر دارد : ۱)مقدار پول موجود در آن ۲) یک کد قفل کننده (ScriptPubKey) که کلید عمومی [۲] و سایر مشخصات کسی که میتواند خروجی را خرج کند مشخص میکند. مثلا در شکل ؟؟ به خروجی شماره ۱ تراکنش سمت چپ دقت کنید. این خروجی دو بیتکوین دارد و کد قفل کننده ScriptPubKey_1، کلید عمومی کسی که میتواند این پول را خرج کند مشخص میکند. این خروجی توسط ورودی شماره ۱ تراکنش سمت راست خرج میشود. هر ورودی سه داده را در بر دارد ۱) هش تراکنشی که میخواهد یکی از خروجی های آن را خرج کند (در این مثال هش تراکنش سمت چپ که با قرمز رنگ مشخص شده است) ۲) شماره خروجی مورد نظر(در این مثال، عدد ۱ که با سبز رنگ مشخص شده است) ۳) یک کد باز کننده قفل که شامل امضای صاحب پول و سایر اثبات های مورد نیاز خروجی است (در این مثال، ScriptSig_1 که با رنگ سرمه ای مشخص شده است) . همچنین دقت کنید که در تراکنش سمت راست ورودی ۲ بیتکوین دارد و مجموع خروجی ها ۱/۹۹ بیتکوین است؛ ۰/۰۱ بیتکوین هم به عنوان کارمزد شبکه در نظر گرفته شده است.



شکل ۲-۱: ساختار یک تراکنش در بیتکوین

توجه کنید که کد قفل کننده و باز کننده قفل باید سازگار باشند مثلا دو عبارت زیر سازگار هستند:

ScriptPubKey: locked with <PubKey_A>
ScriptSig: Signature of A

کد قفل کننده میتواند شروط بیشتر و پیچیده تری هم برای خرج کننده خروجی ایجاب کند. مثلا به کد قفل کننده و بازکننده زیر توجه کنید که به امضای هر دو کاربر A و B نیازدارد. این خروجی مانند یک حساب دو کاربره عمل میکند زیرا برای خرج کردن پول آن تایید هر دو کاربر A و B لازم است.

[۲] public key

ScriptPubKey: locked with <PubKey_A> and <PubKey_B>
ScriptSig: Signature of A and signature of B

قفل زمانی تراکنش های بیتکوین میتوانند شامل جزئیات دیگری مانند قفل زمانی[۳] هم باشند. قفل زمانی به این معنی است که یک تراکنش را پیش از زمان مقرر نمیتوان به بلاکچین ارسال کرد. به طور مثال اگر شما تراکنشی با قفل زمانی December 31 بسازید و آن را پیش از این تاریخ به بلاکچین ارسال کنید، ماینر ها این تراکنش را ثبت نمیکنند و تا روز December 31 صبر کرده و بعد آن را ثبت میکنند.

خروجی های خرج نشده [۴] به خروجی هایی که تاکنون خرج نشده اندUnspent Transaction Output (UTXO)میگویند. ماینر[۵] ها در شبکه بیتکوین دو وظیفه اصلی دارند:

۱. اطمینان حاصل کنند که هیچ خروجی ای بیش از یکبار خرج نمیشود .

۲. بررسی کنند که هر کد باز کننده به درستی کد قفل کننده متناظر را باز میکند.

۲-۲ تراکنش های کانال پرداخت

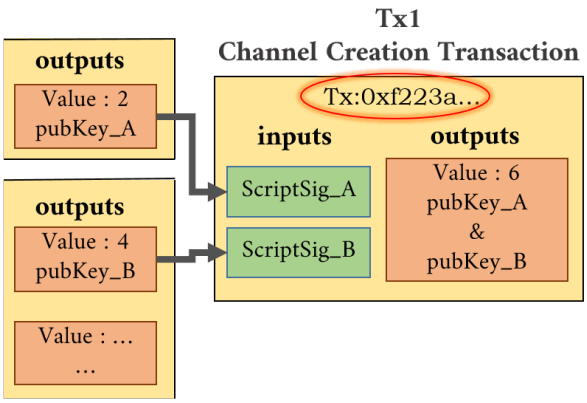
در این بخش نوع ساخت و استفاده از کانال پرداخت های بر مبنای زمان[۶] ها را توضیح میدهیم. کانال پرداخت های رایج در Lightning Network معمولا از نوع punishment based payment channel هستند که جزئیات پیچیده تری نسبت به کانال های بر مبنای زمان دارند. با این وجود چون اصول اولیه و کلیات هر دو این پروتکل ها مشابه هم است، فهم اصول کانال های بر مبنای زمان کافی است. برای خواندن درباره تفاوت این دو پروتکل ایجاد کانال میتوانید به منبع [؟] مراجعه کنید.

۲-۲-۱ ایجاد کانال

همانطور که پیش از این گفته شد، دو کاربر برای ایجاد کانال پرداخت باید یک تراکنش درون بلاکچینی "ایجاد کانال" بسازند. شکل ؟؟ تراکنش Tx1 را نشان میدهد که نمونه ای از یک تراکنش ایجاد کانال

[۳] time lock
[۴] UTXO
[۵] miner
[۶] time based payment channels

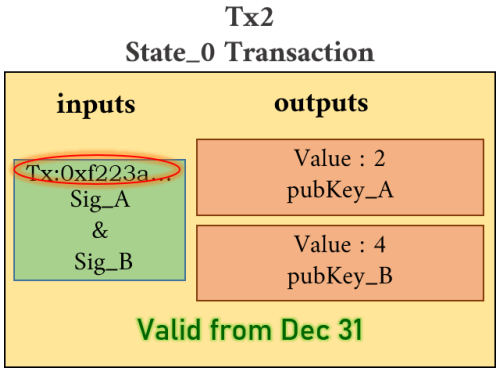
است. A یکی از UTXO هایش به ارزش ۲ بیتکوین و B یکی از UTXO هایش به ارزش ۴ بیتکوین را در کانال سپرده میکنند. خروجی Tx1 یک UTXO دو کاربره با موجودی ۶ است.



شکل ۲-۲: مثالی از یک تراکنش ایجاد کانال پرداخت. این تراکنش درون بلاکچینی است یعنی روی بلاکچین بیتکوین فرستاده میشود.

۲-۲-۲ استفاده از کانال

همزمان یا اندکی پیش از امضای تراکنش درون بلاکچینی Tx1، A و B مشترکا یک تراکنش برون بلاکچینی (Tx2) هم ساخته و هر دو آن امضا میکنند اما آن را روی بلاکچین نمیفرستند، این تراکنش تنها در حافظه محلی A و B ذخیره میشود.

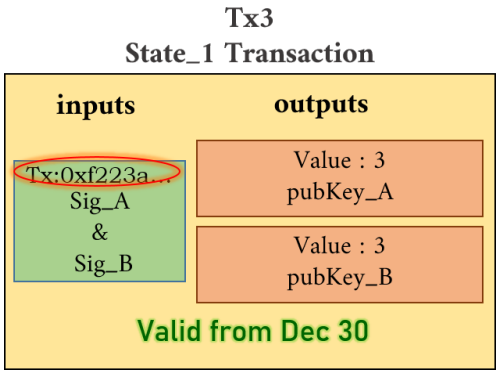


شکل ۲-۳: تراکنش حالت صفر کانال پرداخت. این تراکنش برون بلاکچینی است یعنی توسط A و B مشترکا ساخته و امضا شده و ذخیره می شود ولی تا زمان بسته شدن کانال روی بلاکچین قرار نمیگیرد.

Tx2 تک خروجی تراکنش Tx1 را خرج میکند و ، پول موجود در کانال را به همان نسبت اولیه ۴ – ۲ بین A و B تقسیم میکند. تراکنش Tx2 تنها پس از زمان مقرر قفل زمانی(در این مثال December 31)

بر بلاکچین ثبت میشود. Tx2 در واقع توزیع پول در کانال را در لحظه ایجاد آن یا لحظه صفر نشان میدهد به همین دلیل به آن تراکنش لحظه صفر میگوییم. در صورتی که هیچ تراکنشی بین A و B انجام نگیرد و پس از مدتی یکی از A یا B تصمیم بگیرد کانال را ببندد، آن فرد میتواند تراکنش Tx2 را روی بلاکچین قرار دهد. چون Tx2 پیش از این توسط هر دو A و B امضا شده است پس تراکنش معتبری است و بعد از تاریخ قفل زمانی آن A و B هر کدام میتوانند سهم خود را از کانال پرداخت بگیرند و کانال بسته میشود. اما در عمل A و B کانال پرداخت ساخته اند تا از آن استفاده کنند نه اینکه آن را بلافاصله ببندند پس تراکنش Tx2 عملا هیچگاه استفاده نمیشود. این تراکنش تنها و تنها ساخته میشود تا به هر دو طرف تضمین دهد که اگر طرف دیگر پاسخگو نبود، پول آن ها در کانال قفل نمیماند و هر کدام میتوانند سهم خود را از کانال دریافت کنند.

اکنون با یک مثال توضیح میدهیم که در عمل چگونه از کانال پرداخت ایجاد شده در شکل ؟؟ استفاده میشود. فرض کنید B میخواهد برای A ۱ بیتکوین در کانال پول بریزد. B ترکنش برون بلاکچینی Tx3 نمایش داده شده در شکل ؟؟ را میسازد و امضا میکند و برای A میفرستد. A هم Tx3 را امضا کرده و ذخیره میکند. Tx3 توزیع پول موجود در کانال را به ۳ – ۳ تغییر میدهد. هرگاه هر کدام از A یا B بخواهند این کانال را ببندند کافی است Tx3 را روی بلاکچین بفرستد و در تاریخ December 30 سهم خود از کانال را پس بگیرند.



شکل ۲-۴:

اما یک مشکل وجود دارد؛ چگونه تضمین دهیم که هیچ کدام از طرفین تراکنش قدیمی تر Tx2 را روی بلاکچین قرار نمیدهند تا کانال را با یک توزیع پول قدیمی ببندند؟ این عمل به طور خاص به نفع A است زیرا موجودی A در Tx2 بیشتر از Tx3 است. محدودیت زمانی اعمال شده روی خروجی تراکنش های Tx2 و Tx3 نقش جلوگیری از این عمل را دارد؛ فرض کنید A میخواهد سر B کلاه بگذارد و بدون هیچ اطلاع قبلی تراکنش Tx2 را روی بلاکچین قرار میدهد؛ B با دیدن این عمل، تراکنش Tx3 را به

بلاکچین میفرستد. قفل زمانی Tx3 روز 30 December باز میشود پس این تراکنش یک روز زودتر از Tx2 قابل اجرا است. تراکنش Tx3 در روز 30 December انجام میشود و بعد از آن، تراکنش Tx2 دیگر قابل اجرا نخواهد بود زیرا ورودی آن قبلا توسط Tx3 مصرف شده است.

۲-۳ تراکنش های با واسطه

۲-۴ Lightning Network

۲-۵ الگوریتم های آنلاین

[۷]

[۷] online algorithms

فصل ۳

مدلسازی مسئله

فصل ۴

مفاهیم اولیه

دومین فصل پایان‌نامه به طور معمول به معرفی مفاهیمی می‌پردازد که در پایان‌نامه مورد استفاده قرار می‌گیرند. در این فصل نمونه‌ای از مفاهیم اولیه آورده شده است.

۴-۱ برنامه‌ریزی خطی

در برنامه‌ریزی ریاضی سعی بر بهینه‌سازی (کمینه یا بیشینه کردن) یک تابع هدف با توجه به تعدادی محدودیت است. شکل خاصی از این برنامه‌ریزی که توجه ویژه‌ای به آن در علوم کامپیوتر شده است برنامه‌ریزی خطی می‌باشد. در برنامه‌ریزی خطی به دنبال بهینه کردن یک تابع هدف خطی با توجه به تعدادی محدودیت خطی می‌باشیم. شکل استاندارد یک برنامه‌ریزی خطی به صورت زیر است.

$$\begin{aligned} \text{minimize} \quad & c^T x \\ \text{s.t.} \quad & Ax \geqslant b \\ & x \geqslant 0 \end{aligned} \tag{۴-۱}$$

در روابط فوق، x بردار متغیرها، b, c بردارهای ثابت و A ماتریس ضرایب می‌باشد. به سادگی قابل مشاهده است که رابطه‌ی (؟؟) می‌تواند شکل‌های مختلفی از برنامه‌ریزی خطی را در بر بگیرد. به طور خاص اگر روابط قیدها به حالت $(A'x = b')$ یا در جهت برعکس $(A''x \leqslant b'')$ باشد یا تابع هدف به صورت بیشینه‌سازی باشد. همه‌ی این موارد با تغییر کمی در رابطه‌ی (؟؟) یا اضافه کردن پارامتر و

متغیر جدید قابل مدل کردن می‌باشد. برای مطالعه‌ی بیشتر در مورد برنامه‌ریزی خطی می‌توانید به [؟] مراجعه کنید.

هر برنامه‌ریزی خطی مطرح شده به شکل بالا قابل حل در زمان چندجمله‌ای است [؟، ؟]. روش بیضوی [؟] از این مزیت بهره می‌برد که نیازی به بررسی همه‌ی محدودیت‌ها ندارد. در حقیقت این روش با در اختیار داشتن یک دانای کل جداکننده[۱] می‌تواند جواب بهینه‌ی برنامه‌ریزی خطی را در زمان چندجمله‌ای بدست آورد. دانای کل جداکننده رویه‌ای است که با گرفتن بردار x به عنوان ورودی مشخص می‌کند که آیا x همه‌ی محدودیت‌های برنامه‌ریزی خطی را برآورده می‌سازد یا خیر، در حالت دوم دانای کل جداکننده حداقل یک محدودیت نقض شده را گزارش می‌دهد. این مسئله زمانی کمک کننده خواهد بود که برنامه‌ریزی خطی دارای تعداد نمایی محدودیت باشد اما ساختار ترکیبیاتی محدودیت‌ها امکان ارزیابی امکان‌پذیر بودن جواب مورد نظر را فراهم آورد.

برای هر برنامه‌ریزی خطی می‌توان شکل دوگان آن را نوشت. به برنامه‌ی اصلی، برنامه‌ی اولیه گفته می شود. دوگان رابطه‌ی (؟؟) به صورت زیر می‌باشد:

$$\begin{aligned} \text{maximize} \quad & b^T y \\ \text{s.t.} \quad & A^T y \leqslant c \\ & y \geqslant ۰ \end{aligned} \tag{۲-۴}$$

برنامه‌های اولیه و دوگان به کمک قضایای دوگانی زیر با هم ارتباط دارند.

قضیه‌ی ۴-۱ (قضیه‌ی دوگانی ضعیف) *یک برنامه‌ریزی خطی کمینه‌سازی با تابع هدف $c^T x$ و صورت دوگان آن با تابع هدف $b^T y$ را در نظر بگیرید. برای هر جواب ممکن x برای برنامه‌ی اولیه و جواب ممکن y برای برنامه‌ی دوگان، رابطه‌ی $b^T y \leqslant c^T x$ برقرار است.*

درستی قضیه‌ی بالا به راحتی قابل تصدیق است زیرا $b^T y \leqslant (Ax)^T y = x^T A^T y \leqslant x^T c = c^T x$، برقراری نامساوی‌ها از نامساوی‌های برنامه‌ی اولیه و دوگان حاصل می‌شود. قضیه‌ی قوی دوگانی در [؟] به صورت زیر بیان شده است.

قضیه‌ی ۴-۲ (قضیه‌ی دوگانی قوی) *یک برنامه‌ریزی خطی کمینه‌سازی با تابع هدف $c^T x$ و صورت دوگان آن با تابع هدف $b^T y$ را در نظر بگیرید. اگر برنامه‌ی اولیه یا دوگان دارای جواب بهینه‌ی نامحدود*

[۱] Separation Oracle

باشد، برنامه‌ی متقابل فاقد جواب ممکن است. در غیر این صورت مقدار بهینه‌ی توابع هدف دو برنامه مساوی خواهد بود، به عبارت دیگر جواب x^* برای برنامه ی اولیه و جواب y^* برای برنامه‌ی دوگان وجود خواهد داشت که $c^T x^* = b^T y^*$.

درصورتی مقادیر متغیرها محدود به اعداد صحیح شود به عنوان مثال $x \in \{۰,۱\}^n$ به این شکل از برنامه‌ریزی، برنامه‌ریزی صحیح می‌گوییم. این شکل از برنامه‌ریزی به سادگی قابل بهینه‌سازی نیستند. برداشتن محدودیت صحیح بودن متغیرها، برنامه‌ریزی خطی تعدیل‌شده را نتیجه می‌دهد. بهترین الگوریتم‌ها برای بسیاری از مسائل با گرد کردن جواب برنامه‌ریزی خطی تعدیل شده به مقادیر صحیح یا با بهره‌گیری از ویژگی‌های برنامه‌ریزی خطی (نظیر روش اولیه‌ـ دوگان [؟]) حاصل شده است. دقت کنید که جواب برنامه‌ریزی خطی تعدیل‌شده برای یک مسئله، به عنوان حد پایینی برای جواب بهینه‌ی آن مسئله محسوب می‌گردد.

زمانی که از برنامه‌ریزی خطی تعدیل شده برای حل یا تقریب زدن یک مسئله استفاده می‌شود، گپ صحیح[۲] برنامه‌ریزی خطی معمولاً بیانگر این است که جواب ما تا چه حد می‌تواند مناسب باشد. برای یک مسئله‌ی کمینه‌سازی، گپ صحیح به صورت کوچک‌ترین کران بالای مقدار برنامه‌ریزی خطی تعدیل شده برای نمونه‌ی I تقسیم بر مقدار بهینه برای نمونه‌ی I تعریف می‌شود. گپ صحیح برای مسئله‌ی بیشینه‌سازی به صورت معکوس تقسیم مطرح شده بیان می‌گردد.

۴-۲ الگوریتم‌های تقریبی

بسیاری از مسائل بهینه‌سازی مهم و پایه‌ای انپی‌ـ سخت هستند. بنابراین، با فرض $P \neq NP$ نمی‌توان الگوریتم‌هایی با زمان چندجمله‌ای برای این مسائل ارائه کرد. روش‌های متداول برای برخورد با این مسائل عبارت‌اند از:

- مسئله را فقط برای حالات خاص حل نمود.
- با استفاده از روش‌های جست‌وجوی تمام حالات، مسئله را در زمان غیرچندجمله‌ای حل نمود.
- در زمان چندجمله‌ای، تقریبی از جواب بهینه را به دست آورد.

[۲] Integrality Gap

مسئله	ضریب تقریب
Euclidian TSP	$1+\varepsilon\ (\varepsilon>0)$
Vertex Cover	const c
Set Cover	$\log n$
Coloring	$n^{\delta}\ (\delta<1)$
TSP	∞

جدول ۴-۱: نمونه‌هایی از ضرایب تقریب برای مسائل بهینه‌سازی

در این پایان‌نامه تمرکز بر روی روش سوم یعنی استفاده از الگوریتم‌های تقریبی است. الگوریتم‌های تقریبی قادرند جوابی نزدیک به جواب بهینه را در زمان چندجمله‌ای پیدا کنند.

مسئله‌ی بهینه‌سازی (کمینه‌سازی یا بیشینه‌سازی) P را در نظر بگیرید. فرض کنید هر نمونه از مسئله‌ی P دارای یک مجموعه‌ی ناتهی از جواب‌های ممکن[۳] است. به هر جواب ممکن، یک عدد مثبت به عنوان هزینه (یا وزن) آن نسبت داده شده است. مسئله‌ی P با شرایط فوق یک مسئله‌ی *ان‌پی‌ـ بهینه‌سازی* (NP-Optimization) است،

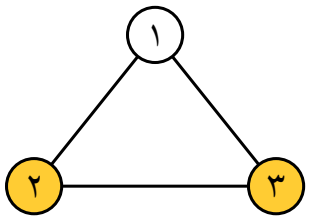
به ازای هر نمونه‌ی I از یک مسئله‌ی ان‌پی‌ـ بهینه‌سازی P، هزینه‌ی جواب بهینه برای I را با $\mathrm{OPT}(I)$ نشان می‌دهیم. همچنین، هزینه‌ی جواب تولیدشده توسط الگوریتم تقریبی بر روی I را با $\mathrm{ALG}(I)$ نشان می‌دهیم.

تعریف ۴-۱ *یک الگوریتم تقریبی برای مسئله‌ی P دارای ضریب تقریب α است اگر برای هر نمونه‌ی I از P:*

$$\max\left\{\frac{ALG(I)}{OPT(I)}, \frac{OPT(I)}{ALG(I)}\right\} \leqslant \alpha.$$

یک الگوریتم تقریبی با ضریب تقریب α، یک *الگوریتم αـ تقریبی* نامیده می‌شود. نمونه‌هایی از ضرایب تقریب متداول برای مسائل بهینه‌سازی در جدول ؟؟ آمده است.

[۳] feasible



شکل ۴-۱: گراف G و یک پوشش رأسی برای آن

۴-۳ پوشش رأسی

به عنوان اولین مسئله از مجموعه مسائل بهینه‌سازی، در این بخش به بررسی مسئله‌ی پوشش رأسی می‌پرازیم. این مسئله به صورت زیر تعریف می‌شود.

مسئله‌ی ۴-۱ (پوشش رأسی) *گراف $G = (V, E)$ و تابع هزینه‌ی $w : V \to \mathbb{R}^+$ داده شده است. زیرمجموعه‌ی $C \subseteq V$ با حداقل هزینه را بیابید طوری که به ازای هر یال $uv \in E$، حداقل یکی از دو رأس u و v در مجموعه‌ی C باشد.*

شکل ؟؟ نمونه‌ای از یک پوشش رأسی را نشان می‌دهد. در زیر یک الگوریتم حریصانه برای مسئله‌ی پوشش رأسی غیروزن‌دار ارائه شده است.

الگوریتم ۱ پوشش رأسی حریصانه

۱: قرار بده $C = \emptyset$
۲: تا وقتی E تهی نیست:
۳: یال دل‌خواه $uv \in E$ را انتخاب کن
۴: $C \leftarrow C \cup \{u, v\}$
۵: تمام یال‌های واقع بر u یا v را از E حذف کن
۶: C را برگردان

به سادگی می‌توان مشاهده نمود که خروجی الگوریتم ؟؟ یک پوشش رأسی است. در ادامه نشان خواهیم داد که اندازه‌ی پوشش رأسی تولیدشده توسط الگوریتم حداکثر دو برابر اندازه‌ی پوشش رأسی کمینه است.

قضیه‌ی ۴-۳ $\mathrm{OPT} \leqslant |C| \leqslant ۲\,\mathrm{OPT}$.

اثبات. از آن جایی که C یک پوشش رأسی است، نامساوی سمت چپ بدیهی است. فرض کنید M مجموعه‌ی تمام یال‌هایی باشد که توسط الگوریتم انتخاب شده‌اند. از آن جایی که هیچ دو یالی در M دارای رأس مشترک نیستند، هر پوشش رأسی (از جمله پوشش رأسی بهینه) باید حداقل یک رأس از هر یال موجود در M را بپوشاند. بنابراین

$$|M| \leqslant \mathrm{OPT}\,.$$

از طرفی می‌دانیم $|C| = ۲|M|$. در نتیجه

$$|C| = ۲|M| \leqslant ۲\,\mathrm{OPT}\,.$$

□

بنا بر قضیه‌ی ؟؟، الگوریتم ؟؟ یک الگوریتم ۲ـتقریبی است. مثال زیر نشان می‌دهد که ضریب تقریب ۲ برای این الگوریتم محکم است. گراف دو بخشی کامل $K_{n,n}$ را در نظر بگیرید. پوشش رأسی تولیدشده توسط الگوریتم حریصانه بر روی این گراف شامل تمامی $۲n$ رأس گراف خواهد بود، در صورتی که پوشش رأسی بهینه شامل نصف این تعداد، یعنی n رأس است.

فصل ۵

کارهای پیشین

در این فصل کارهای پیشین انجام‌شده روی مسئله به تفصیل توضیح داده می‌شود.

فصل ۶

نتایج جدید

در این فصل نتایج جدید به‌دست‌آمده در پایان‌نامه توضیح داده می‌شود. در صورت نیاز می‌توان نتایج جدید را در قالب چند فصل ارائه نمود. همچنین در صورت وجود پیاده‌سازی، بهتر است نتایج پیاده‌سازی را در فصل مستقلی پس از این فصل قرار داد.

فصل ۷

نتیجه‌گیری

در این فصل، ضمن جمع‌بندی نتایج جدید ارائه‌شده در پایان‌نامه، مسائل باز باقی‌مانده و همچنین پیش‌نهادهایی برای ادامه‌ی کار ارائه می‌شوند.

فصل ۸

مدلسازی مسئله

فصل ۹

نحوه‌ی نگارش

سلامممممممممممممممممممممس در این فصل نکات کلی در مورد نگارش پایان‌نامه به اختصار توضیح داده می‌شود.

۹-۱ پرونده‌ها

پرونده‌ی اصلی پایان‌نامه‌ی شما `thesis.tex` نام دارد. به ازای هر فصل از پایان‌نامه، یک پرونده در شاخه‌ی `chapters` ایجاد نموده و نام آن را در پرونده‌ی `thesis.tex` (در قسمت فصل‌ها) درج نمایید. پیش از شروع به نگارش پایان‌نامه، بهتر است پرونده‌ی `front/info.tex` را باز نموده و مشخصات پایان‌نامه را در آن تغییر دهید.

۹-۲ عبارات ریاضی

برای درج عبارات ریاضی در داخل متن از $...$ و برای درج عبارات ریاضی در یک خط مجزا از $$...$$ استفاده کنید. برای مثال $\sum_{k=۰}^{n} \binom{n}{k} = ۲^n$ در داخل متن و عبارت زیر

$$\sum_{k=۰}^{n} \binom{n}{k} = ۲^n$$

در یک خط مجزا درج شده است. همان‌طور که در بالا می‌بینید، نمایش یک عبارت یکسان در دو حالت درون‌خط و بیرون‌خط می‌تواند متفاوت باشد. دقت کنید که تمامی عبارات ریاضی، از جمله متغیرهای تک‌حرفی مانند x و y باید در محیط ریاضی یعنی محصور درون علامت $ باشند.

۹-۳ علائم ریاضی پرکاربرد

برخی علائم ریاضی پرکاربرد در زیر فهرست شده‌اند.

- مجموعه‌های اعداد: $\mathbb{N}, \mathbb{Z}, \mathbb{Z}^+, \mathbb{Q}, \mathbb{R}, \mathbb{C}$
- مجموعه: $\{۱, ۲, ۳\}$
- دنباله: $\langle ۱, ۲, ۳ \rangle$
- سقف و کف: $\lceil x \rceil, \lfloor x \rfloor$
- اندازه و متمم: $|A|, \overline{A}$
- همنهشتی: $a \overset{n}{\equiv} ۱$ یا (پیمانه‌ی n) $a \equiv ۱$
- ضرب و تقسیم: $\times, \cdot, \div$
- سه‌نقطه بین کاما: $۱, ۲, \ldots, n$
- سه‌نقطه بین عملگر: $۱ + ۲ + \cdots + n$
- کسر و ترکیب: $\frac{n}{k}, \binom{n}{k}$
- اجتماع و اشتراک: $A \cup (B \cap C)$
- عملگرهای منطقی: $\neg p \vee (q \wedge r)$
- پیکان‌ها: $\to, \Rightarrow, \leftarrow, \Leftarrow, \leftrightarrow, \Leftrightarrow$
- عملگرهای مقایسه‌ای: $\neq, \leqslant, \nleqslant, \geqslant, \ngeqslant$

- عملگرهای مجموعه‌ای: $\in, \notin, \setminus, \subset, \subseteq, \subsetneq, \supset, \supseteq, \supsetneq$
- جمع و ضرب چندتایی: $\sum_{i=۱}^{n} a_i, \prod_{i=۱}^{n} a_i$
- اجتماع و اشتراک چندتایی: $\bigcup_{i=۱}^{n} A_i, \bigcap_{i=۱}^{n} A_i$
- برخی نمادها: $\infty, \emptyset, \forall, \exists, \triangle, \angle, \ell, \equiv, \therefore$

۹-۴ لیست‌ها

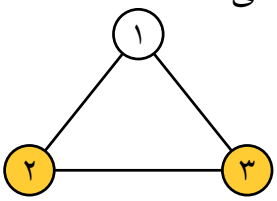
برای ایجاد یک لیست می‌توانید از محیط‌های «فقرات» و «شمارش» همانند زیر استفاده کنید.

- مورد اول
- مورد دوم
- مورد سوم

۱. مورد اول
۲. مورد دوم
۳. مورد سوم

۹-۵ درج شکل

یکی از روش‌های مناسب برای ایجاد شکل استفاده از نرم‌افزار LaTeX Draw و سپس درج خروجی آن به صورت یک فایل `tex` درون متن با استفاده از دستور `fig` یا `centerfig` است. شکل ؟؟ نمونه‌ای از اشکال ایجادشده با این ابزار را نشان می‌دهد.



شکل ۹-۱: یک گراف و پوشش رأسی آن

همچنین می‌توانید با استفاده از نرم‌افزار Ipe شکل‌های خود را مستقیما به صورت pdf ایجاد نموده و آن‌ها را با دستورات `img` یا `centerimg` درون متن درج کنید. برای نمونه، شکل ؟؟ را ببینید.

عملگر	عملیات
<	کوچک‌تر
>	بزرگ‌تر
==	مساوی
<>	نامساوی

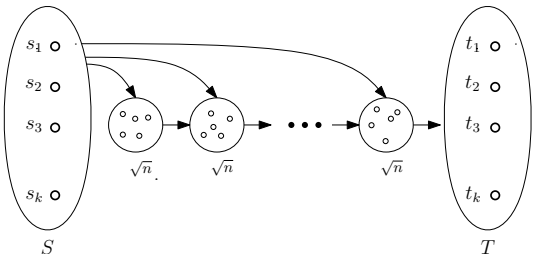
جدول ۹-۱: عملگرهای مقایسه‌ای

۹-۶ درج جدول

برای درج جدول می‌توانید با استفاده از دستور «جدول» جدول را ایجاد کرده و سپس با دستور «لوح» آن را درون متن درج کنید. برای نمونه جدول ؟؟ را ببینید.

۹-۷ درج الگوریتم

برای درج الگوریتم می‌توانید از محیط «الگوریتم» همانند زیر استفاده کنید.



شکل ۹-۲: یک گراف جهت‌دار بدون دور

الگوریتم ۲ پوشش رأسی حریصانه

ورودی: گراف $G = (V, E)$

خروجی: یک پوشش رأسی از G

۱: قرار بده $C = \emptyset$

۲: تا وقتی E تهی نیست:

۳: یال دل‌خواه $uv \in E$ را انتخاب کن

۴: رأس‌های u و v را به C اضافه کن

۵: تمام یال‌های واقع بر u یا v را از E حذف کن

۶: C را برگردان

۹-۸ محیط‌های ویژه

برای درج مثال‌ها، قضیه‌ها، لم‌ها و نتیجه‌ها به ترتیب از محیط‌های «مثال»، «قضیه»، «لم» و «نتیجه» استفاده کنید. برای درج اثبات قضیه‌ها و لم‌ها از محیط «اثبات» استفاده کنید.

تعریف‌های داخل متن را با استفاده از دستور «مهم» به صورت **تیره** نشان دهید. تعریف‌های پایه‌ای‌تر را درون محیط «تعریف» قرار دهید.

تعریف ۹-۱ (اصل لانه‌کبوتری) *اگر $n+1$ یا بیش‌تر کبوتر درون n لانه قرار گیرند، آن‌گاه لانه‌ای وجود دارد که شامل حداقل دو کبوتر است.*

فصل ۱۰

برخی نکات نگارشی

این فصل حاوی برخی نکات ابتدایی ولی بسیار مهم در نگارش متون فارسی است. نکات گردآوری‌شده در این فصل به هیچ وجه کامل نیست، ولی دربردارنده‌ی حداقل مواردی است که رعایت آن‌ها در نگارش پایان‌نامه ضروری به نظر می‌رسد.

۱۰-۱ فاصله‌گذاری

۱. علائم سجاوندی مانند نقطه، ویرگول، دونقطه، نقطه‌ویرگول، علامت سؤال، و علامت تعجب (. ، : ؛ ؟ !) بدون فاصله از کلمه‌ی پیشین خود نوشته می‌شوند، ولی بعد از آن‌ها باید یک فاصله قرار گیرد. مانند: من، تو، او.

۲. علامت‌های پرانتز، آکولاد، کروشه، نقل قول و نظایر آن‌ها بدون فاصله با عبارات داخل خود نوشته می‌شوند، ولی با عبارات اطراف خود یک فاصله دارند. مانند: (این عبارت) یا آن عبارت.

۳. دو کلمه‌ی متوالی در یک جمله همواره با یک فاصله از هم جدا می‌شوند، ولی اجزای یک کلمه‌ی مرکب باید با نیم‌فاصله[۱] از هم جدا شوند. مانند: کلاسِ درس، محبت‌آمیز، دوبخشی.

[۱] «نیم‌فاصله» فاصله‌ای مجازی است که در عین جدا کردن اجزای یک کلمه‌ی مرکب از یک‌دیگر، آن‌ها را نزدیک به هم نگه می‌دارد. معمولاً برای تولید این نوع فاصله در صفحه‌کلیدهای استاندارد از ترکیب Shift+Space استفاده می‌شود.

۱۰-۲ شکل حروف

۱. در متون فارسی به جای حروف «ك» و «ي» عربی باید از حروف «ک» و «ی» فارسی استفاده شود. همچنین به جای اعداد عربی مانند ٥ و ٦ باید از اعداد فارسی مانند ۵ و ۶ استفاده نمود. برای این کار، توصیه می‌شود صفحه‌کلید فارسی استاندارد[۲] را بر روی سیستم خود نصب کنید.

۲. عبارات نقل‌قول‌شده یا مؤکد باید درون علامت نقل قولِ «» قرار گیرند، نه "". مانند: «کشور ایران».

۳. کسره‌ی اضافه‌ی بعد از «ه» غیرملفوظ به صورت «ه‌ی» نوشته می‌شود، نه «هٔ». مانند: خانه‌ی علی، دنباله‌ی فیبوناچی.

تبصره: اگر «ه» ملفوظ باشد، نیاز به «ی» ندارد. مانند: فرمانده دلیر، پادشه خوبان.

۴. پایه‌های همزه در کلمات، همیشه «ئـ» است، مانند: مسئله و مسئول، مگر در مواردی که همزه ساکن است که در این صورت باید متناسب با اعراب حرف پیش از خود نوشته شود. مانند: رأس، مؤمن.

۱۰-۳ جدانویسی

۱. اجزای فعل‌های مرکب با فاصله از یک‌دیگر نوشته می‌شوند، مانند: تحریر کردن، به سر آمدن.

۲. علامت استمرار، «می»، توسط نیم‌فاصله از جزء بعدی فعل جدا می‌شود. مانند: می‌رود، می‌توانیم.

۳. شناسه‌های «ام»، «ای»، «ایم»، «اید» و «اند» توسط نیم‌فاصله، و شناسه‌ی «است» توسط فاصله از کلمه‌ی پیش از خود جدا می‌شوند. مانند: گفته‌ام، گفته‌ای، گفته است.

۴. علامت جمع «ها» توسط نیم‌فاصله از کلمه‌ی پیش از خود جدا می‌شود. مانند: این‌ها، کتاب‌ها.

۵. «به» همیشه جدا از کلمه‌ی بعد از خود نوشته می‌شود، مانند: به نام و به آن‌ها، مگر در مواردی که «بـ» صفت یا فعل ساخته است. مانند: بسزا، ببینم.

[۲] صفحه‌کلید فارسی استاندارد برای ویندوز، تهیه‌شده توسط بهنام اسفهبد

۶. «به» همواره با فاصله از کلمه‌ی بعد از خود نوشته می‌شود، مگر در مواردی که «به» جزئی از یک اسم یا صفت مرکب است. مانند: تناظر یک‌به‌یک، سفر به تاریخ.

۱۰-۴ جدانویسی مرجح

۱. اجزای اسم‌ها، صفت‌ها، و قیدهای مرکب توسط نیم‌فاصله از یک‌دیگر جدا می‌شوند. مانند: دانش‌جو، کتاب‌خانه، گفت‌وگو، آن‌گاه، دل‌پذیر.

تبصره: اجزای منتهی به «هاء ملفوظ» را می‌توان از این قانون مستثنی کرد. مانند: راهنما، رهبر.

۲. علامت صفت برتری، «تر»، و علامت صفت برترین، «ترین»، توسط نیم‌فاصله از کلمه‌ی پیش از خود جدا می‌شوند. مانند: بیش‌تر، کم‌ترین.

تبصره: کلمات «بهتر» و «بهترین» را می‌توان از این قاعده مستثنی نمود.

۳. پیشوندها و پسوندهای جامد، چسبیده به کلمه‌ی پیش یا پس از خود نوشته می‌شوند. مانند: همسر، دانشکده، دانشگاه.

تبصره: در مواردی که خواندن کلمه دچار اشکال می‌شود، می‌توان پسوند یا پیشوند را جدا کرد. مانند: هم‌میهن، هم‌ارزی.

۴. ضمیرهای متصل چسبیده به کلمه‌ی پیش از خود نوشته می‌شوند. مانند: کتابم، نامت، کلامشان.

مسئله‌ی ۱۰-۱ *گراف غیر جهت‌دار $G = (V, E)$ به همراه m رأس مشخص $d_1, d_2, \ldots, d_m$ از V به عنوان انبار و m تابع وزن $w_1, w_2, \ldots, w_m : V \times V \to \mathbb{R}^+$ داده شده است. در هر یک از انبارها یک عامل (وسیله‌ی نقلیه) قرار دارد. هدف یافتن m دور است که از $d_1, d_2, \ldots, d_m$ شروع شده و اجتماع آن‌ها تمام رأس‌های گراف را بپوشاند طوری که مجموع هزینه‌ی این دورها کمینه شود. هزینه‌ی دور iاُم با تابع w_i اندازه‌گیری می‌شود.*

در صورت همگن مسئله، هزینه‌ی پیمایش یال‌ها برای همه‌ی عوامل یکسان است و در گونه‌ی ناهمگن، این هزینه برای عوامل مختلف می‌تواند متفاوت باشد. از آن جایی که صورت ناهمگن مسئله کم‌تر مورد توجه قرار گرفته است، در این تحقیق سعی شده است که تمرکز بر روی این گونه از مسئله

باشد. همچنین علاوه بر دورهای ناهمگن، درخت‌ها و مسیرهای ناهمگن نیز در این پایان‌نامه مورد بررسی قرار خواهند گرفت.

مسئله‌ی مسیریابی وسایل نقلیه کاربردهای بسیار گسترده‌ای در حوزه‌ی حمل و نقل دارد. برای نخستین بار این مسئله برای مسیریابی تانکرهای سوخت‌رسان مطرح شد [؟]. اما امروزه با پیشرفت‌های گسترده‌ای که در زمینه‌ی تکنولوژی روی داده است از راه‌حل‌های این مسئله در امور روزمره از جمله سیستم توزیع محصولات، تحویل نامه، جمع‌آوری زباله‌های خانگی و غیره استفاده می‌شود. در نظر گرفتن فرض ناهمگن بودن هم با توجه به اینکه معمولاً عوامل توزیع در یک سیستم، یکسان نیستند و تفاوت‌هایی در میزان مصرف سوخت و غیره دارند، راه‌حل‌های مناسب‌تری برای مسائل این حوزه می‌تواند ارائه دهد. گونه‌های مختلفی از مسائل مسیریابی وسایل نقلیه در [؟، ؟، ؟] بیان شده است.

همان‌طور که ذکر شد مسئله‌ی مسیریابی وسایل نقلیه‌ی ناهمگن صورت عمومی مسئله‌ی فروشنده دوره‌گرد می‌باشد. مسئله‌ی فروشنده‌ی دوره‌گرد در حوزه‌ی مسائل ان‌پی‌ـ‌سخت[۳] قرار می‌گیرد و با فرض $P \neq NP$ الگوریتم دقیق با زمان چندجمله‌ای برای آن وجود ندارد. بنابراین برای حل کارای این مسائل از الگوریتم‌های تقریبی[۴] استفاده می‌شود.

مسئله‌ی فروشنده‌ی دوره‌گرد در حالتی که تنها یک فروشنده در گراف حضور داشته باشد، دو الگوریتم تقریبی معروف دارد. در الگوریتم اول با دو برابر کردن درخت پوشای کمینه[۵] و میانبر کردن[۶] دورهای بدست آمده، الگوریتمی با ضریب تقریب ۲ ارائه می‌شود. در الگوریتم دوم که متعلق به کریستوفایدز[۷] [؟] است، به کمک ساخت دور اویلری[۸] بر روی اجتماع یال‌های درخت پوشای کمینه و یال‌های تطابق کامل کمینه[۹] از گره‌های درجه‌ی فرد همان درخت، و میانبر کردن این دور، ضریب تقریب ۱/۵ ارائه می‌شود. با گذشت حدود ۴۰ سال از ارائه‌ی این الگوریتم، تا کنون ضریب تقریب بهتری برای این مسئله پیدا نشده است.

اخیراً با بهره‌گیری از روش کریستوفایدز و بسط آن برای مسئله‌ی فروشنده‌ی دوره‌گرد چندگانه‌ی همگن (در این حالت از مسئله تعداد فروشنده‌ها در گراف بیش از یکی است و هزینه‌ی پیمایش یال‌ها برای همه‌ی عوامل یکسان است) ضریب تقریب ۱/۵ ارائه شده است [؟]. در روش مطرح شده بعد از

[۳] NP-hard
[۴] Approximation Algorithm
[۵] Minimum Spanning Tree
[۶] Shortcut
[۷] Christofides
[۸] Eulerian Cycle
[۹] Minimum Perfect Matching

به دست آوردن درخت‌های پوشای کمینه برای هر انبار، به جای استفاده از روش دو برابر کردن یال‌ها، روش کریستوفایدز اعمال می‌شود. به راحتی می‌توان نشان داد که صرف اعمال الگوریتم کریستوفایدز به هر یک از درخت‌های بدست آمده، ضریب تقریب ۱/۵ را بدست نمی‌دهد. بنابراین در روش مذکور، الگوریتم کریستوفایدز روی کل جنگل بدست آمده اعمال می‌شود. نشان داده شده است که با استفاده از یک سیاست جایگزینی مناسب بین یال‌هایی که در جنگل کمینه، موجود هستند و آن‌هایی که در این مجموعه حضور ندارند و اعمال کریستوفایدز روی این جنگل‌ها، می‌توان جوابی تولید کرد که بدتر از ۱/۵ برابر جواب بهینه نباشد.

همان‌طور که گفته شد نسخه‌ی ناهمگن این مسئله کمتر مورد توجه قرار گرفته است. در گونه‌ی ناهمگن، بیش از یک عامل (فروشنده) در اختیار داریم که در شروع، هر یک از آن‌ها در گره‌های مجزایی که با عنوان انبار معرفی می‌شوند قرار دارند و هزینه‌ی پیمایش یال‌ها برای هریک از عوامل می‌تواند متفاوت از سایر عامل‌ها باشد. در صورتی که تعداد انبارها m فرض شود از جمله کارهای انجام شده در این مورد ارائه ضریب تقریب $۴m$ به کمک حل برنامه‌ریزی خطی تعدیل شده[۱۰] و ساخت درخت پوشای کمینه [؟]، ضریب تقریب $۱/۵m$ به کمک حل تعدیل برنامه‌ریزی خطی با روش بیضی[۱۱] و اعمال الگوریتم کریستوفایدز [؟] و ضریب تقریب ۲ به کمک راه حل اولیه‌ـ دوگان[۱۲] می‌باشد، روش اولیه‌ـ دوگان تنها برای حالتی که دو عامل وجود دارد و هزینه‌ی پیمایش یال‌ها برای یک عامل بیشتر از عامل دیگر باشد مطرح شده است [؟].

در برنامه‌ریزی ریاضی سعی بر بهینه‌سازی (کمینه یا بیشینه کردن) یک تابع هدف با توجه به تعدادی محدودیت است. شکل خاصی از این برنامه‌ریزی که توجه ویژه‌ای به آن در علوم کامپیوتر شده است برنامه‌ریزی خطی می‌باشد. در برنامه‌ریزی خطی به دنبال بهینه کردن یک تابع هدف خطی با توجه به تعدادی محدودیت خطی می‌باشیم. شکل استاندارد یک برنامه‌ریزی خطی به صورت زیر است.

$$\begin{aligned} \text{minimize} \quad & c^T x \\ \text{s.t.} \quad & Ax \geqslant b \\ & x \geqslant ۰ \end{aligned} \qquad (۱-۱۰)$$

در روابط فوق، x بردار متغیرها، b, c بردارهای ثابت و A ماتریس ضرایب می‌باشد. به سادگی قابل مشاهده است که رابطه‌ی (؟؟) می‌تواند شکل‌های مختلفی از برنامه‌ریزی خطی را در بر بگیرد. به طور

[۱۰] Linear Programming Relaxation
[۱۱] Ellipsoid Method
[۱۲] Primal-Dual

خاص اگر روابط قیدها به حالت $(A'x = b')$ یا در جهت برعکس $(A''x \leqslant b'')$ باشد یا تابع هدف به صورت بیشینه‌سازی باشد. همه‌ی این موارد با تغییر کمی در رابطه‌ی (؟؟) یا اضافه کردن پارامتر و متغیر جدید قابل مدل کردن می‌باشد. برای مطالعه‌ی بیشتر در مورد برنامه‌ریزی خطی می‌توانید به [؟] مراجعه کنید.

هر برنامه‌ریزی خطی مطرح شده به شکل بالا قابل حل در زمان چندجمله‌ای است [؟، ؟]. روش بیضوی [؟] از این مزیت بهره می‌برد که نیازی به بررسی همه‌ی محدودیت‌ها ندارد. در حقیقت این روش با در اختیار داشتن یک دانای کل جداکننده[۱۳] می‌تواند جواب بهینه‌ی برنامه‌ریزی خطی را در زمان چندجمله‌ای بدست آورد. دانای کل جداکننده رویه‌ای است که با گرفتن بردار x به عنوان ورودی مشخص می‌کند که آیا x همه‌ی محدودیت‌های برنامه‌ریزی خطی را برآورده می‌سازد یا خیر، در حالت دوم دانای کل جداکننده حداقل یک محدودیت نقض شده را گزارش می‌دهد. این مسئله زمانی کمک کننده خواهد بود که برنامه‌ریزی خطی دارای تعداد نمایی محدودیت باشد اما ساختار ترکیبیاتی محدودیت‌ها امکان ارزیابی امکان‌پذیر بودن جواب مورد نظر را فراهم آورد.

برای هر برنامه‌ریزی خطی می‌توان شکل دوگان آن را نوشت. به برنامه‌ی اصلی، برنامه‌ی اولیه گفته می شود. دوگان رابطه‌ی (؟؟) به صورت زیر می‌باشد:

$$\begin{aligned} \text{maximize} \quad & b^T y \\ \text{s.t.} \quad & A^T y \leqslant c \\ & y \geqslant 0 \end{aligned} \tag{۲-۱۰}$$

برنامه‌های اولیه و دوگان به کمک قضایای دوگانی زیر با هم ارتباط دارند.

قضیه‌ی ۱۰-۱ (قضیه‌ی دوگانی ضعیف) *یک برنامه‌ریزی خطی کمینه‌سازی با تابع هدف $c^T x$ و صورت دوگان آن با تابع هدف $b^T y$ را در نظر بگیرید. برای هر جواب ممکن x برای برنامه‌ی اولیه و جواب ممکن y برای برنامه‌ی دوگان، رابطه‌ی $b^T y \leqslant c^T x$ برقرار است.*

درستی قضیه‌ی بالا به راحتی قابل تصدیق است زیرا $b^T y \leqslant (Ax)^T y = x^T A^T y \leqslant x^T c = c^T x$، برقراری نامساوی‌ها از نامساوی‌های برنامه‌ی اولیه و دوگان حاصل می‌شود. قضیه‌ی قوی دوگانی در [؟] به صورت زیر بیان شده است.

[۱۳] Separation Oracle

قضیه‌ی ۱۰-۲ (قضیه‌ی دوگانی قوی) *یک برنامه‌ریزی خطی کمینه‌سازی با تابع هدف $c^T x$ و صورت دوگان آن با تابع هدف $b^T y$ را در نظر بگیرید. اگر برنامه‌ی اولیه یا دوگان دارای جواب بهینه‌ی نامحدود باشد، برنامه‌ی متقابل فاقد جواب ممکن است. در غیر این صورت مقدار بهینه‌ی توابع هدف دو برنامه مساوی خواهد بود، به عبارت دیگر جواب x^* برای برنامه ی اولیه و جواب y^* برای برنامه‌ی دوگان وجود خواهد داشت که $c^T x^* = b^T y^*$.*

درصورتی مقادیر متغیرها محدود به اعداد صحیح شود به عنوان مثال $x \in \{۰, ۱\}^n$ به این شکل از برنامه‌ریزی، برنامه‌ریزی صحیح می‌گوییم. این شکل از برنامه‌ریزی به سادگی قابل بهینه‌سازی نیستند. برداشتن محدودیت صحیح بودن متغیرها، برنامه‌ریزی خطی تعدیل‌شده را نتیجه می‌دهد. بهترین الگوریتم‌ها برای بسیاری از مسائل با گرد کردن جواب برنامه‌ریزی خطی تعدیل شده به مقادیر صحیح یا با بهره‌گیری از ویژگی‌های برنامه‌ریزی خطی (نظیر روش اولیه‌ـ دوگان [؟]) حاصل شده است. دقت کنید که جواب برنامه‌ریزی خطی تعدیل‌شده برای یک مسئله، به عنوان حد پایینی برای جواب بهینه‌ی آن مسئله محسوب می‌گردد.

زمانی که از برنامه‌ریزی خطی تعدیل شده برای حل یا تقریب زدن یک مسئله استفاده می‌شود، گپ صحیح[۱۴] برنامه‌ریزی خطی معمولاً بیانگر این است که جواب ما تا چه حد می‌تواند مناسب باشد. برای یک مسئله‌ی کمینه‌سازی، گپ صحیح به صورت کوچک‌ترین کران بالای مقدار برنامه‌ریزی خطی تعدیل شده برای نمونه‌ی I تقسیم بر مقدار بهینه برای نمونه‌ی I تعریف می‌شود. گپ صحیح برای مسئله‌ی بیشینه‌سازی به صورت معکوس تقسیم مطرح شده بیان می‌گردد.

بسیاری از مسائل بهینه‌سازی مهم و پایه‌ای انپی‌ـ سخت هستند. بنابراین، با فرض $P \neq NP$ نمی‌توان الگوریتم‌هایی با زمان چندجمله‌ای برای این مسائل ارائه کرد. روش‌های متداول برای برخورد با این مسائل عبارت‌اند از:

- مسئله را فقط برای حالات خاص حل نمود.
- با استفاده از روش‌های جست‌وجوی تمام حالات، مسئله را در زمان غیرچندجمله‌ای حل نمود.
- در زمان چندجمله‌ای، تقریبی از جواب بهینه را به دست آورد.

[۱۴] Integrality Gap

مسئله	ضریب تقریب
Euclidian TSP	$1+\varepsilon\ (\varepsilon>0)$
Vertex Cover	const c
Set Cover	$\log n$
Coloring	$n^{\delta}\ (\delta<1)$
TSP	∞

جدول ۱۰-۱: نمونه‌هایی از ضرایب تقریب برای مسائل بهینه‌سازی

در این پایان‌نامه تمرکز بر روی روش سوم یعنی استفاده از الگوریتم‌های تقریبی است. الگوریتم‌های تقریبی قادرند جوابی نزدیک به جواب بهینه را در زمان چندجمله‌ای پیدا کنند.

مسئله‌ی بهینه‌سازی (کمینه‌سازی یا بیشینه‌سازی) P را در نظر بگیرید. فرض کنید هر نمونه از مسئله‌ی P دارای یک مجموعه‌ی ناتهی از جواب‌های ممکن[۱۵] است. به هر جواب ممکن، یک عدد مثبت به عنوان هزینه (یا وزن) آن نسبت داده شده است. مسئله‌ی P با شرایط فوق یک مسئله‌ی *ان‌پی‌ـ بهینه‌سازی* (NP-Optimization) است،

به ازای هر نمونه‌ی I از یک مسئله‌ی ان‌پی‌ـ بهینه‌سازی P، هزینه‌ی جواب بهینه برای I را با $\mathrm{OPT}(I)$ نشان می‌دهیم. همچنین، هزینه‌ی جواب تولیدشده توسط الگوریتم تقریبی بر روی I را با $\mathrm{ALG}(I)$ نشان می‌دهیم.

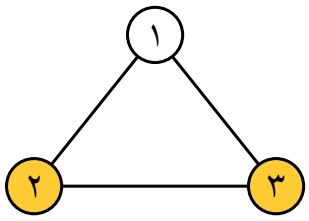
تعریف ۱۰-۱ *یک الگوریتم تقریبی برای مسئله‌ی P دارای ضریب تقریب α است اگر برای هر نمونه‌ی I از P:*

$$\max\left\{\frac{ALG(I)}{OPT(I)}, \frac{OPT(I)}{ALG(I)}\right\} \leqslant \alpha.$$

یک الگوریتم تقریبی با ضریب تقریب α، یک *الگوریتم αـ تقریبی* نامیده می‌شود. نمونه‌هایی از ضرایب تقریب متداول برای مسائل بهینه‌سازی در جدول ؟؟ آمده است.

به عنوان اولین مسئله از مجموعه مسائل بهینه‌سازی، در این بخش به بررسی مسئله‌ی پوشش رأسی می‌پرازیم. این مسئله به صورت زیر تعریف می‌شود.

[۱۵] feasible



شکل ۱۰-۱: گراف G و یک پوشش رأسی برای آن

مسئله‌ی ۱۰-۲ (پوشش رأسی) *گراف $G = (V, E)$ و تابع هزینه‌ی $w : V \to \mathbb{R}^+$ داده شده است. زیرمجموعه‌ی $C \subseteq V$ با حداقل هزینه را بیابید طوری که به ازای هر یال $uv \in E$، حداقل یکی از دو رأس u و v در مجموعه‌ی C باشد.*

شکل ؟؟ نمونه‌ای از یک پوشش رأسی را نشان می‌دهد. در زیر یک الگوریتم حریصانه برای مسئله‌ی پوشش رأسی غیروزن‌دار ارائه شده است.

الگوریتم ۳ پوشش رأسی حریصانه

۱: قرار بده $C = \emptyset$
۲: تا وقتی E تهی نیست:
۳: یال دل‌خواه $uv \in E$ را انتخاب کن
۴: $C \leftarrow C \cup \{u, v\}$
۵: تمام یال‌های واقع بر u یا v را از E حذف کن
۶: C را برگردان

به سادگی می‌توان مشاهده نمود که خروجی الگوریتم ؟؟ یک پوشش رأسی است. در ادامه نشان خواهیم داد که اندازه‌ی پوشش رأسی تولیدشده توسط الگوریتم حداکثر دو برابر اندازه‌ی پوشش رأسی کمینه است.

قضیه‌ی ۱۰-۳ $\mathrm{OPT} \leqslant |C| \leqslant ۲\,\mathrm{OPT}$.

اثبات. از آن جایی که C یک پوشش رأسی است، نامساوی سمت چپ بدیهی است. فرض کنید M مجموعه‌ی تمام یال‌هایی باشد که توسط الگوریتم انتخاب شده‌اند. از آن جایی که هیچ دو یالی در M

دارای رأس مشترک نیستند، هر پوشش رأسی (از جمله پوشش رأسی بهینه) باید حداقل یک رأس از هر یال موجود در M را بپوشاند. بنابراین

$$|M| \leqslant \mathrm{OPT}\,.$$

از طرفی می‌دانیم $|C| = ۲|M|$. در نتیجه

$$|C| = ۲|M| \leqslant ۲\,\mathrm{OPT}\,.$$

□

بنا بر قضیه‌ی ؟؟، الگوریتم ؟؟ یک الگوریتم ۲ـ تقریبی است. مثال زیر نشان می‌دهد که ضریب تقریب ۲ برای این الگوریتم محکم است. گراف دو بخشی کامل $K_{n,n}$ را در نظر بگیرید. پوشش رأسی تولیدشده توسط الگوریتم حریصانه بر روی این گراف شامل تمامی $۲n$ رأس گراف خواهد بود، در صورتی که پوشش رأسی بهینه شامل نصف این تعداد، یعنی n رأس است.

پیوست آ

مطالب تکمیلی

پیوست‌های خود را در صورت وجود می‌توانید در این قسمت قرار دهید.

Bibliography

[1] J. Poon and T. Dryja. The bitcoin lightning network: Scalable off-chain instant payments. https://lightning.network/lightning-network-paper.pdf, 2015.

[2] Raiden network. https://raiden.network/, 2017.

[3] C. Decker. Lightning network research; topology, datasets. https://github.com/lnresearch/topology. Accessed: 2022-04-01.

[4] Lightning network search and analysis engine.

واژه‌نامه

الف

ابتکاری ..... heuristic

ارزش ..... worth

ارضاپذیری ..... satisfiability

استراتژی ..... strategy

ائتلاف ..... coalition

ب

بارگذاری ..... loading

بازی ..... game

برچسب ..... label

برنامه‌ریزی خطی ..... linear programming

برنامه‌ریزی صحیح ..... integer programming

بسته‌بندی ..... packing

بهترین پاسخ ..... best response

بیشینه ..... maximum

پ

پالت ..... pallet

پایداری ..... robustness

پشتیبان ..... support

پوسته‌ی محدب ..... convex hull

پوش بالایی ..... upper envelope

پوششی ..... covering

ت

تبدیل تصویری ..... projective transformation

تعادل ..... equlibrium

تعدیل ..... relaxation

تقاطع ..... intersection

تقسیم‌بندی ..... partition

تکاملی ..... evolutionary

توزیع‌شده ..... distributed

ج

جست‌وجوی جامع ..... brute-force

جست‌وجوی عمق‌اول ..... Depth-First Search

جعبه ..... bin

چ

چاله ..... sink

ح

حرکت ..... action

خ

خودخواهانه ..... selfish

خوشه ..... clique

د

دودویی ..... binary

دوگان ..... dual

دو ماتریسی ..... bimatrix

ر

رأس ..... vertex

رفتار ..... behaviour

رنگ‌آمیزی ..... coloring

ز

زمان‌بندی ..... scheduling

زیست‌شناسی ..... biology

س

ساختی ..... constructive

سود ..... pay off, utility

ش

شبه‌چندجمله‌ای ..... quasi-polynomial

شبه‌مقعر ..... quasi-concave

ص

صوری ..... formal

ع

عاقل ..... rational

عامل‌ـ‌محور ..... agent-based

عمل ..... action

غ

غائب ..... missing

غیرمتمرکز ..... decentralized

غیرمعمول ..... degenerate

ق

قابل انتقال ..... transferable

قاموسی ..... lexicographically

قوی ..... strong

ک

کمینه ..... minimum

م

مجموع زیرمجموعه‌ها ..... subset sum
مجموعه ..... set
محور ..... pivot
مختلط ..... mixed
مخفی ..... hidden
مستوی ..... affine
مسطح ..... planar
منطقی ..... reasonable
موازی ..... parallel

ن

نتیجه‌ی نهایی ..... outcome
نش ..... Nash
نقطه‌ثابت ..... fixed point
نگارخانه‌ی هنر ..... art gallery
نگهبان ..... gaurd
نمایه ..... profile
نوبتی ..... round-robin

و

وجه ..... facet

هـ

هزینه‌ی آشوب ..... price of anarchy (POA)
هزینه‌ی اجتماعی ..... social cost
هزینه‌ی پایداری ..... price of stability (POS)

ی

یال ..... edge
یکریختی ..... isomorphism

Abstract

We present a standard template for typesetting theses in Persian. The template is based on the XePersian package for the LaTeX typesetting system. This write-up shows a sample usage of this template.

Keywords: Thesis, Typesetting, Template, XePersian

Sharif University of Technology

Department of Computer Engineering

M.Sc. Thesis

A Standard Template for Typesetting Theses in Persian

By:

Hamid Zarrabi-Zadeh

Supervisor:

Dr. Supervisor

September 2020