



دانشگاه صنعتی شریف  
دانشکده‌ی مهندسی کامپیوتر

پایان‌نامه‌ی کارشناسی ارشد  
مهندسی نرم‌افزار

عنوان:

## قالب استاندارد برای نگارش پایان‌نامه‌ها

نگارش:

حمید ضربابی‌زاده

استاد راهنما:

استاد راهنمای پروژه

شهریور ۱۳۹۹

سلام افلا

## چکیده

امروزه استفاده از شبکه کانال های پرداخت<sup>۱</sup> 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 برای هم بفرستند و چون این تراکنش ها به صورت محلی<sup>۲</sup> و برون بلاکچینی<sup>۳</sup> انجام میشود 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.

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 تا حد امکان مقدار سپرده کافی در کانال خود از همان ابتدا قرار دهند. از طرفی کاربران مایل نیستند بیش از مقدار نیاز واقعی خود در کانال پول ذخیره کنند 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 وقتی شبکه ای از این کانال های پرداخت ایجاد میشود، تصمیمات متعدد دیگری هنگام نگهداری یک کانال باید گرفته شود.

یک کاربر میتواند به تعداد دلخواه با کاربران متفاوت دیگر، کانال پرداخت ایجاد کند و به این صورت از اتصال کاربران به هم توسط کانال های پرداخت، شبکه ای از کانال های پرداخت<sup>۴</sup> ایجاد میشود. 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 میتوانند در صورت امکان با استفاده از

---

<sup>۱</sup> payment channel

<sup>۲</sup> local

<sup>۳</sup> بدون مراجعه به بلاکچین off-chain transaction

<sup>۴</sup> payment channel network

سایر نود های شبکه به عنوان واسطه، تراکنش خود را در شبکه مسیریابی و ارسال کنند. بنابراین نود های موجود در شبکه میتوانند در دو نقش کاربر (فرستنده یا گیرنده) یا سرویس دهنده (واسطه) ایفای نقش کنند. نود های واسطه با انتقال تراکنش های کاربران کارمزد دریافت میکنند اما از طرفی قبول حریصانه تمام تراکنش ها توسط واسطه ها ممکن است کانال های آنها را نامتعادل کند، به این معنی که موجودی یک طرف کانال صفر شود و همه سپرده دست طرف دیگر باشد که این امر مانع انتقال تراکنش های بعدی از جهت بدون پول میشود. بدین صورت نود های شبکه کانال های پرداخت با یک تصمیم گیری آنلاین دیگر هم روبرو هستند؛ اینکه کدام یک تراکنش های کاربران را انتقال دهند. بنابراین همانطور که توضیح داده شد میبینیم که نود های شبکه کانال های پرداخت چه هنگام ایجاد کانال و چه بعدا باید مدام تصمیمات آنلاینی در خصوص نگهداری کانال خود بگیرند.

طراحی یک الگوریتم بهینه آنلاین که درباره مدیریت سپرده های نو 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 این مسأله تصمیم گیری به همراه تضمین تئوری کارکرد آن ارائه میدهیم. همچنین 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

# فهرست مطالب

۹	۱ مقدمه
۱۲	۱-۱ اهمیت موضوع
۱۲	۲-۱ دست آورد های تحقیق
۱۴	۳-۱ ساختار پایان نامه
۱۵	۲ پیش نیازها
۱۵	۱-۲ نحوه کار یک کانال پرداخت در Lightning Network
۱۵	۲-۲ نحوه کار تراکش های با واسطه
۱۵	۳-۲ الگوریتم های آنلاین
۱۶	۳ مفاهیم اولیه
۱۶	۱-۳ برنامه ریزی خطی
۱۸	۲-۳ الگوریتم های تقریبی
۲۰	۳-۳ پوشش رأسی
۲۲	۴ کارهای پیشین
۲۳	۵ نتایج جدید

## ۶ نتیجه‌گیری ۲۴

## ۷ نحوه‌ی نگارش ۲۵

۱-۷ پرونده‌ها ۲۵

۲-۷ عبارات ریاضی ۲۵

۳-۷ علائم ریاضی پرکاربرد ۲۶

۴-۷ لیست‌ها ۲۷

۵-۷ درج شکل ۲۷

۶-۷ درج جدول ۲۸

۷-۷ درج الگوریتم ۲۸

۸-۷ محیط‌های ویژه ۲۹

## ۸ برخی نکات نگارشی ۳۰

۱-۸ فاصله‌گذاری ۳۰

۲-۸ شکل حروف ۳۱

۳-۸ جدانویسی ۳۱

۴-۸ جدانویسی مرجع ۳۲

## آ مطالب تکمیلی ۴۰

## فهرست تصاویر

۲۰	۱-۳	گراف $G$ و یک پوشش رأسی برای آن
۲۷	۱-۷	یک گراف و پوشش رأسی آن
۲۸	۲-۷	یک گراف جهت‌دار بدون دور
۳۸	۱-۸	گراف $G$ و یک پوشش رأسی برای آن

## فهرست جداول

- ۱-۳ نمونه‌هایی از ضرایب تقریب برای مسائل بهینه‌سازی . . . . . ۱۹
- ۱-۷ عملگرهای مقایسه‌ای . . . . . ۲۸
- ۱-۸ نمونه‌هایی از ضرایب تقریب برای مسائل بهینه‌سازی . . . . . ۳۷



# فصل ۱

## مقدمه

بلاکچین‌هایی همچون بیتکوین به دلیل ماهیت توزیع شده والگوریتم اجماع پیچیده و وقت‌گیری که دارند با مشکل عدم مقیاس پذیری روبرو هستند. 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، تایید شدن تراکنش‌ها در بلاکچین‌ها معمولاً حداقل چند دقیقه به طول می‌انجامد. یکی از مورد استقبال‌ترین راه‌حل‌هایی که برای حل مشکل مقیاس‌ناپذیری و کندی بلاکچین ارائه شده است استفاده از شبکه کانال‌های پرداخت<sup>۱</sup> است. شبکه کانال‌های پرداخت اولین بار با پیاده‌سازی Lightning Network روی بلاکچین بیتکوین معرفی شد. [۱] بعد از شبکه کانال‌های پرداخت Raiden هم با الهام از Lightning Network بر بلاکچین اتریوم<sup>۲</sup> توسعه داده شد. [۲] کاربران میتوانند با ارسال یک تراکنش ایجاد کانال<sup>۳</sup> 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<sup>۴</sup> با کارمزد بسیار اندک و بفرستند. مبادله امضاها دیجیتال برای حفظ امنیت مالی طرفین الزامی است؛ در واقع در اینجا طرفین کانال پرداخت به هم اعتماد ندارند و پیش فرض اولیه این است که هر کدام از طرفین ممکن است تلاش در دزدیدن پول دیگری کند؛ بنابراین پروتکل با استفاده از اصول رمزنگاری به نحوی طراحی شده است که از این امر جلوگیری کند. به عنوان

<sup>۱</sup> payment channel network

<sup>۲</sup> Ethereum

<sup>۳</sup> channel creation

<sup>۴</sup> instant

مثالی از نحوه کارکرد کانال پرداخت، دو کاربر A و B را در نظر بگیرید که هنگام ایجاد کانال به ترتیب ۲ و ۴ بیتکوین در کانال به سپرده میگذارند. فرض کنید A میخواهد ۱ بیتکوین به B بدهد؛ این دو نفر امضاهای دیجیتالی مبادله میکنند که وضعیت کانال را از ۲ - ۴ به ۵ - ۱ تغییر میدهد اما این تغییر در این مرحله بر بلاکچین منعکس نمیشود. در نهایت وقتی طرفین تصمیم به بستن کانال گرفتند آخرین امضاهای مبادله شده را که نمایانگر موجودی نهایی طرفین است را روی بلاکچین قرار میدهند و هر کدام از طرفین سهم خودش را دریافت میکند و کانال بسته میشود. مثلاً در مثال بالا ممکن است پس از انجام رشته ای از تراکنش ها موجودی طرفین ۳ - ۳ باشد، پس هر کدام از A و B ۳ بیتکوین دریافت کرده و کانال پرداخت بسته میشود. بدین ترتیب با ارسال تنها دو تراکنش on-chain (تراکنش های ایجاد و بستن کانال) میتوان تعداد زیادی تراکنش off-chain ارسال کرد.

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. مثلاً 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 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 میشوند و یا هر دو برگشت میخورند. معمولاً فرد واسطه یعنی B مقداری کارمزد از A میگیرد اما این کارمزد در برابر کارمزد های شبکه بیتکوین بسیار ناچیز است و صرفاً نقش ایجاد انگیزه برای واسطه ها را دارد. البته گاهی نود های واسطه ممکن است برخی تراکنش ها را به دلایلی رد کنند. مثلاً 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. مثلاً 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؛ موجودی آنها ۶ - ۰ خواهد بود و حالا 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<sup>۵</sup> میگوییم. کانال های نامتعادل برای کاربران عادی مطلوب نیستند زیرا امکان ایجاد تراکنش از یک سمت

depleted channel<sup>۵</sup>

کانال را به کل از بین میبرند اما کانال های نامتعادل به خصوص برای نود های واسطه یا سرویس دهنده<sup>۶</sup> بسیار مشکل ساز هستند. نود های سرویس دهنده نود هایی هستند که 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 نود های واسطه 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 صرفاً با تعدادی تراکنش برون بلاکچینی توزیع سپرده ها را در کانال نامتعادل تغییر میدهد و به نسبت روش قبل ارزان تر است. این عمل به این صورت انجام میگردد که طرفین کانال یک مسیر دوم از کانال های پرداخت بین خودشان پیدا میکنند و با جابجایی پول در یک حلقه فردی که همه پول را در کانال نامتعادل به دست دارد برای فرد دیگر اندکی پول میریزد و معادل آن پول را در یک کانال دیگر از آن نود دریافت میکنند.

از آنجاییکه هر دو روش بالا هزینه بر هستند و محدودیت هایی را اعمال میکنند، تصمیم گیری بر سر اینکه چه زمانی کدام یک از آنها انجام گیرد تصمیم سختی میتواند باشد. همچنین توجه کنید که به عنوان یک کاربر یا یک نود واسطه، معمولاً نود اطلاعات دقیقی از تراکنش های آینده ندارد و در نتیجه نود ها باید سیاست تصمیم گیری ای را اتخاذ کنند که پیش فرض خاصی درباره توزیع تراکنش ها ندارد بنابراین سیاست تصمیم گیری باید آنلاین باشد.

هدف این پایان نامه این است که سیاست آنلاین طراحی کند یک 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:

<sup>۶</sup> service provider

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 در واقع ایجاد بستری ارزان و سریع برای انجام تراکنش های کوچک و روزانه<sup>۷</sup> 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 معمولاً هزینه اصلاح یک کانال نامتعادل بسیار زیاد است. از طرف دیگر، مدیریت نادرست کانال ها توسط سرویس دهنده ها باعث کاهش سود آنها میشود. اگر سود ناشی از انتقال تراکنش ها در برابر هزینه های نگهداری کانال قابل توجه نباشد، سرویس دهنده ها انگیزه ای برای ماندن در شبکه کانال های پرداخت ندارند. این امر برای کارکرد شبکه کانال های پرداخت بسیار خطرناک است زیرا وجود واسطه ها برای انتقال تراکنش کاربرانی که کانال مستقیم ندارند الزامی است. در نتیجه ارائه الگوریتمی که این مسئله را با کمترین فروض و به صورت بهینه حل کند، بسیار ارزشمند است.

## ۲-۱ 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

<sup>۷</sup> micro payment

میکنیم طوری که مسأله نهایی تا حد خوبی بیشتر پیچیدگی های کانال های پرداخت در دنیای واقعی را در خود دارد. این دو زیر مسأله به شرح زیر هستند:

۱. زیر مسأله ۱ (کانال یکطرفه همیشه پذیرنده<sup>۸</sup>) در این زیر مسأله یک کانال پرداخت را در نظر میگیریم که منحصرأ یکی از طرفین کانال تراکنش میفرستد و جهت تراکنش ها همیشه یکطرفه است. همچنین فرض میکنیم تمام تراکنش ها باید پذیرفته شوند بنابراین هرگاه طرفین کانال موجودی کافی برای انتقال تراکنشی نداشته باشند، باید با شارژ کردن درون بلاکچینی کانال خود را شارژ کنند تا بتوانند تمام تراکنش ها را عبور دهند. همچنین در این مدل متعادل کردن برون بلاکچینی مجاز نیست. برای این مدل الگوریتمی آنلاین با ضریب رقابتی  $2 + \frac{\sqrt{5}-1}{2}$  ارائه میدهم و اثبات میکنیم که این بهترین ضریب رقابتی ای ست که یک الگوریتم آنلاین میتواند به آن دست یابد.

۲. زیر مسأله ۲ (کانال یکطرفه مجاز به رد تراکنش<sup>۱۰</sup>) 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. پذیرفتن تراکنش ها پاداش کارمزد را به همراه دارد. مشابه مدل قبل متعادل کردن برون بلاکچینی مجاز نیست. برای این مدل الگوریتم آنلاین با ضریب رقابتی  $2 + \frac{\sqrt{5}-1}{2}$  ارائه میدهم و اثبات میکنیم که این الگوریتم بهینه است.

۳. مسأله اصلی (کانال دوطرفه<sup>۱۱</sup>) 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 ضریب رقابتی  $7 + \log C$  طراحی میکنیم. (  $C$  یک عدد ثابت است که بستگی به ویژگی های گراف شبکه کانال های پرداخت دارد و مثلاً در Lightning Network حدوداً برابر ۴ است ). همچنین به عنوان کران پایین نشان میدهم که هیچ الگوریتم آنلاینی با ضریب 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

<sup>۸</sup>Unidirectional stream without rejection

<sup>۹</sup>competitive ratio 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 میدهم.

<sup>۱۰</sup>Unidirectional stream with rejection

<sup>۱۱</sup>Bidirectional stream

مورد استفاده قرار میگیرد.

دقت کنید که در این بخش مسأله به نحوی بیان شد که همواره یکی از طرفین کانال پرداخت سرویس دهنده باشد اما در؟؟ توضیح میدهیم که چگونه باید پارامترهای مسأله را تغییر داد تا الگوریتم ها بتوانند برای مدیریت کانال دو کاربر عادی که قصد انتقال تراکنش سایرین را ندارند و فقط تراکنش های بین خودشان را میخواهند مدیریت کنند، مورد استفاده قرار گیرد.

## ۳-۱ ساختار پایان نامه

این پایان نامه شامل پنج فصل است. فصل دوم دربرگیرنده تعاریف اولیه مرتبط با پایان نامه است. در فصل سوم مسأله دوره های ناهمگن و کارهای مرتبطی که در این زمینه انجام شده به تفصیل بیان می گردد. در فصل چهارم نتایج جدیدی که در این پایان نامه به دست آمده ارائه می گردد. در این فصل، مسأله درخت های ناهمگن در چهار شکل مختلف مورد بررسی قرار می گیرد. سپس نگاهی کوتاه به مسأله مسیرهای ناهمگن خواهیم داشت. در انتها با تغییر تابع هدف، به حل مسأله کمینه کردن حداکثر اندازه درخت ها می پردازیم. فصل پنجم به نتیجه گیری و پیشنهادهایی برای کارهای آتی خواهد پرداخت.

## فصل ۲

### مفاهیم اولیه

۱-۲ نحوه کار یک کانال پرداخت

۲-۲ نحوه کار تراکنش های با واسطه در شبکه کانال های پرداخت

۳-۲ الگوریتم های آنلاین

۱

## فصل ۳

# 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.

### ۱-۳ 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.

$$\text{minimize } c^T x \quad (1-3)$$

$$\text{s.t. } Ax \geq b$$

$$x \geq 0$$

در روابط فوق،  $x$  بردار متغیرها،  $b, c$  بردارهای ثابت و  $A$  ماتریس ضرایب می‌باشد. به سادگی قابل مشاهده است که رابطه‌ی (۱-۸) 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  $(A''x \leq b'')$  باشد یا تابع هدف به صورت بیشینه‌سازی باشد. همه‌ی این موارد با تغییر کمی در رابطه‌ی (۱-۸) یا اضافه کردن پارامتر و



متغیر جدید قابل مدل کردن می باشد. برای مطالعه‌ی بیشتر در مورد برنامه‌ریزی خطی می‌توانید به [۴] مراجعه کنید.

هر برنامه‌ریزی خطی مطرح شده به شکل بالا قابل حل در زمان چندجمله‌ای است [۴، ۵]. روش بیضوی [۴] از این مزیت بهره می‌برد که نیازی به بررسی همه‌ی محدودیت‌ها ندارد. در حقیقت این روش با در اختیار داشتن یک دانای کل جداکننده<sup>۱</sup> 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 می‌شود. دوگان رابطه‌ی (۸-۱) به صورت زیر می‌باشد:

$$\begin{aligned} \text{maximize} \quad & b^T y \\ \text{s.t.} \quad & A^T y \leq c \\ & y \geq 0 \end{aligned} \quad (2-3)$$

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

**قضیه‌ی ۱-۳ (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  $b^T y \leq c^T x$  برقرار است.

درستی قضیه‌ی بالا به راحتی قابل تصدیق است زیرا  $b^T y \leq (Ax)^T y = x^T A^T y \leq x^T c = c^T x$ ، برقراری نامساوی‌ها از نامساوی‌های برنامه‌ی اولیه و دوگان حاصل می‌شود. قضیه‌ی قوی دوگانی در [۴] به صورت زیر بیان شده است.

**قضیه‌ی ۲-۳ (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

<sup>۱</sup>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 برنامه‌ی اولیه و جواب  $y^*$  برای برنامه‌ی دوگان وجود خواهد داشت که  $c^T x^* = b^T y^*$ .

در صورتی مقادیر متغیرها محدود به اعداد صحیح شود به عنوان مثال  $x \in \{0, 1\}^n$  به این شکل از برنامه‌ریزی، برنامه‌ریزی صحیح می‌گوییم. این شکل از برنامه‌ریزی به سادگی قابل بهینه‌سازی نیستند. برداشتن محدودیت صحیح بودن متغیرها، برنامه‌ریزی خطی تعدیل شده 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 اولیه - دوگان [؟]) حاصل شده است. دقت کنید که جواب برنامه‌ریزی خطی تعدیل شده 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<sup>۲</sup> 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.

## ۲-۳ الگوریتم‌های تقریبی

بسیاری از مسائل بهینه‌سازی مهم و پایه‌ای ان‌پی-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.

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

جدول ۳-۱: 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<sup>۳</sup> 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 *ان‌پی*-بهینه‌سازی (NP-Optimization) است،

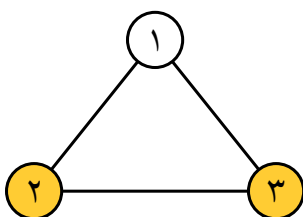
به ازای هر نمونه‌ی  $I$  از یک مسئله‌ی *ان‌پی*-بهینه‌سازی  $P$ ، هزینه‌ی جواب بهینه برای  $I$  را با  $OPT(I)$  نشان می‌دهیم. همچنین، هزینه‌ی جواب تولیدشده توسط الگوریتم تقریبی بر روی  $I$  را با  $ALG(I)$  نشان می‌دهیم.

**تعریف ۳-۱** یک الگوریتم تقریبی برای مسئله‌ی  $P$  دارای ضریب تقریب  $\alpha$  است اگر برای هر نمونه‌ی  $I$  از  $P$ :

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

یک الگوریتم تقریبی با ضریب تقریب  $\alpha$ ، یک الگوریتم  $\alpha$ -تقریبی نامیده می‌شود. نمونه‌هایی از ضرایب تقریب متداول برای مسائل بهینه‌سازی در جدول ۳-۸ آمده است.

<sup>۳</sup>feasible

شکل ۳-۱: گراف  $G$  و یک پوشش رأسی برای آن

### ۳-۳ پوشش رأسی

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

**مسئله ۳-۱ (پوشش رأسی)** گراف  $G = (V, E)$  و تابع هزینه  $w : V \rightarrow \mathbb{R}^+$  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 ۳-۸ نمونه‌ای از یک پوشش رأسی را نشان می‌دهد. در زیر یک الگوریتم حریصانه برای مسئله 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 ۳ یک پوشش رأسی است. در ادامه نشان خواهیم داد که اندازه‌ی پوشش رأسی تولیدشده توسط الگوریتم حداکثر دو برابر اندازه‌ی پوشش رأسی کمینه است.

قضیه ۳-۳.  $\text{OPT} \leq |C| \leq 2 \text{OPT}$ .

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

$$|M| \leq \text{OPT}.$$

از طرفی می‌دانیم  $|C| = 2|M|$ . در نتیجه

$$|C| = 2|M| \leq 2 \text{OPT}.$$

□

بنا بر قضیه ۳-۸، الگوریتم ۳ یک الگوریتم ۲-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  $2n$  رأس گراف خواهد بود، در صورتی که پوشش رأسی بهینه شامل نصف این تعداد، یعنی  $n$  رأس است.

## فصل ۴

# کارهای پیشین

در این فصل کارهای پیشین انجام شده روی مسئله به تفصیل توضیح داده می شود.

## فصل ۵

### نتایج جدید

در این فصل نتایج جدید به دست آمده در پایان نامه توضیح داده می شود. در صورت نیاز می توان نتایج جدید را در قالب چند فصل ارائه نمود. همچنین در صورت وجود پیاده سازی، بهتر است نتایج پیاده سازی را در فصل مستقلی پس از این فصل قرار داد.

## فصل ۶

### نتیجه‌گیری

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



## فصل ۷

# نحوه‌ی نگارش

سلامممممممممممممممممممممممم در این فصل نکات کلی در مورد نگارش پایان نامه به اختصار توضیح داده می شود.

۱-۷ 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.

## ۷-۲ عبارات ریاضی

برای درج عبارات ریاضی در داخل متن از  $\$... \$$  و برای درج عبارات ریاضی در یک خط مجزا از  $\$... \$$  استفاده کنید. برای مثال  $\sum_{k=0}^n \binom{n}{k} = 2^n$  در داخل متن و عبارت زیر

$$\sum_{k=0}^n \binom{n}{k} = 2^n$$

در یک خط مجزا درج شده است. همان‌طور که در بالا می‌بینید، نمایش یک عبارت یکسان در دو حالت درون خط و بیرون خط 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.

## ۳-۷ 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:  $\{1, 2, 3\}$

- دنباله:  $\langle 1, 2, 3 \rangle$

- سقف و کف:  $\lceil x \rceil, \lfloor x \rfloor$

- اندازه و متمم:  $|A|, \overline{A}$

- همنهشتی:  $a \equiv 1 \pmod{n}$  یا  $a \equiv 1 \pmod{n}$  (پیمانه‌ی  $n$ )

- ضرب و تقسیم:  $\times, \cdot, \div$

- سه‌نقطه بین کاما:  $1, 2, \dots, n$

- سه‌نقطه بین عملگر:  $1 + 2 + \dots + n$

- 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:  $\rightarrow, \Rightarrow, \leftarrow, \Leftarrow, \leftrightarrow, \Leftrightarrow$

- عملگرهای مقایسه‌ای:  $\neq, \leq, \not\leq, \geq, \not\geq$

• عملگرهای مجموعه‌ای:  $\in, \notin, \setminus, \subset, \subseteq, \subsetneq, \supset, \supseteq, \supsetneq$

• جمع و ضرب چندتایی:  $\sum_{i=1}^n a_i, \prod_{i=1}^n a_i$

• اجتماع و اشتراک چندتایی:  $\bigcup_{i=1}^n A_i, \bigcap_{i=1}^n A_i$

• برخی نمادها:  $\infty, \emptyset, \forall, \exists, \Delta, \angle, \ell, \equiv, \therefore$

## ۴-۷ لیست‌ها

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

• مورد اول

۱. مورد اول

• مورد دوم

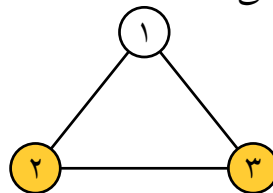
۲. مورد دوم

• مورد سوم

۳. مورد سوم

## ۵-۷ 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 ۷-۱ نمونه‌ای از اشکال ایجادشده با این ابزار را نشان می‌دهد.



شکل ۷-۱: یک گراف و پوشش رأسی آن

همچنین می‌توانید با استفاده از نرم‌افزار Ipe شکل‌های خود را مستقیماً به صورت pdf ایجاد نموده و آن‌ها را با دستورات img یا centerimg درون متن درج کنید. برای نمونه، شکل ۷-۲ را ببینید.

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

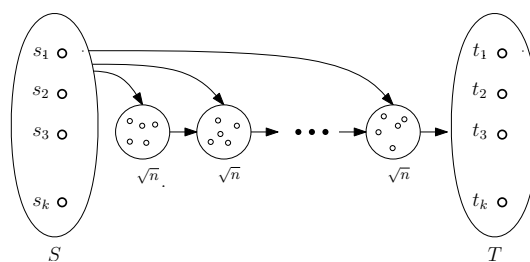
جدول ۷-۱: عملگرهای مقایسه‌ای

## ۶-۷ 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 ۷-۱ را ببینید.

## ۷-۷ درج الگوریتم

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



شکل ۷-۲: 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

---

## ۸-۷ 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 ۷-۱ (اصل لانه کبوتری) اگر  $n+1$  یا بیش‌تر کبوتر درون  $n$  لانه قرار گیرند، آن‌گاه لانه‌ای وجود دارد که شامل حداقل دو کبوتر است.

## فصل ۸

### برخی نکات نگارشی

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

#### ۸-۱ 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<sup>۱</sup> از هم جدا شوند. مانند: کلاس درس، محبت‌آمیز، دوبخشی.

<sup>۱</sup> «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.

## ۸-۲ 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 ۵ و ۶ باید از اعداد فارسی مانند ۵ و ۶ استفاده نمود. برای این کار، توصیه می‌شود صفحه کلید فارسی استاندارد<sup>۲</sup> را بر روی سیستم خود نصب کنید.
۲. عبارات نقل قول شده یا مؤکد باید درون علامت نقل قول «» قرار گیرند، نه «». مانند: «کشور ایران».
۳. کسره‌ی اضافی بعد از «ه» غیرملفوظ به صورت «ه‌ی» نوشته می‌شود، نه «ه‌ة». مانند: خانه‌ی علی، دنباله‌ی فیوناچی.
- تبصره: اگر «ه» ملفوظ باشد، نیاز به «ی» ندارد. مانند: فرمانده دلیر، پادشه خوبان.
۴. پایه‌های همزه در کلمات، همیشه «ئ» 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.

## ۸-۳ 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 «ب» صفت یا فعل ساخته است. مانند: بسزا، ببینم.

<sup>۲</sup> صفحه کلید فارسی استاندارد برای ویندوز، تهیه شده توسط بهنام اسفهد

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.

## ۴-۸ جدانویسی مرجح

۱. اجزای اسم‌ها، صفت‌ها، و قیده‌های 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 «بهر» 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 ۸-۱** گراف غیر جهت‌دار  $G = (V, E)$  به همراه  $m$  رأس مشخص  $d_1, d_2, \dots, d_m$  از  $V$  به عنوان انبار و  $m$  تابع وزن  $w_1, w_2, \dots, w_m : V \times V \rightarrow \mathbb{R}^+$  داده شده است. در هر یک از انبارها یک عامل (وسیله‌ی نقلیه) قرار دارد. هدف یافتن  $m$  دور است که از  $d_1, d_2, \dots, d_m$  شروع شده و اجتماع آن‌ها تمام رأس‌های گراف را بپوشاند طوری که مجموع هزینه‌ی این دورها کمینه شود. هزینه‌ی دور  $i$ ام 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 [۱]. 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 [۱، ۲، ۳] بیان شده است.

همان‌طور که ذکر شد مسئله‌ی مسیریابی وسایل نقلیه‌ی ناهمگن صورت عمومی مسئله‌ی فروشنده دوره‌گرد می‌باشد. مسئله‌ی فروشنده‌ی دوره‌گرد در حوزه‌ی مسائل ان‌پی-سخت<sup>۲</sup> 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<sup>۴</sup> 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<sup>۵</sup> و میانبر کردن<sup>۶</sup> دورهای بدست آمده، الگوریتمی با ضریب تقریب ۲ ارائه می‌شود. در الگوریتم دوم که متعلق به کریستوفایدز<sup>۷</sup> [۱] است، به کمک ساخت دور اویلری<sup>۸</sup> بر روی اجتماع یال‌های درخت پوشای کمینه و یال‌های تطابق کامل کمینه<sup>۹</sup> 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 [۱]. در روش مطرح شده بعد از

NP-hard<sup>۳</sup>Approximation Algorithm<sup>۴</sup>Minimum Spanning Tree<sup>۵</sup>Shortcut<sup>۶</sup>Christofides<sup>۷</sup>Eulerian Cycle<sup>۸</sup>Minimum Perfect Matching<sup>۹</sup>

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  $1/5$  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  $1/5$  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 هر یک 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  $4m$  به کمک حل برنامه‌ریزی خطی تعدیل شده<sup>۱۰</sup> و ساخت درخت پوشای کمینه [؟]، ضریب تقریب  $1/5m$  به کمک حل تعدیل برنامه‌ریزی خطی با روش بیضی<sup>۱۱</sup> و اعمال الگوریتم کریستوفایدز [؟] و ضریب تقریب ۲ به کمک راه حل اولیه-دوگان<sup>۱۲</sup> می‌باشد، روش اولیه-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.

$$\text{minimize } c^T x \quad (1-8)$$

$$\text{s.t. } Ax \geq b$$

$$x \geq 0$$

در روابط فوق،  $x$  بردار متغیرها،  $b, c$  بردارهای ثابت و  $A$  ماتریس ضرایب می‌باشد. به سادگی قابل مشاهده است که رابطه‌ی (۱-۸) می‌تواند شکل‌های مختلفی از برنامه‌ریزی خطی را در بر بگیرد. به طور

<sup>۱۰</sup>Linear Programming Relaxation

<sup>۱۱</sup>Ellipsoid Method

<sup>۱۲</sup>Primal-Dual

خاص اگر روابط قیدها به حالت  $(A'x = b')$  یا در جهت برعکس  $(A''x \leq b'')$  باشد یا تابع هدف به صورت بیشینه‌سازی باشد. همه‌ی این موارد با تغییر کمی در رابطه‌ی (۸-۱) یا اضافه کردن پارامتر و متغیر جدید قابل مدل کردن می‌باشد. برای مطالعه‌ی بیشتر در مورد برنامه‌ریزی خطی می‌توانید به [۹] مراجعه کنید.

هر برنامه‌ریزی خطی مطرح شده به شکل بالا قابل حل در زمان چندجمله‌ای است [۹، ؟]. روش بیضوی [۹] از این مزیت بهره می‌برد که نیازی به بررسی همه‌ی محدودیت‌ها ندارد. در حقیقت این روش با در اختیار داشتن یک دانای کل جداکننده<sup>۱۳</sup> 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 نامایی 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 می‌شود. دوگان رابطه‌ی (۸-۱) به صورت زیر می‌باشد:

$$\begin{aligned} \text{maximize} \quad & b^T y \\ \text{s.t.} \quad & A^T y \leq c \\ & y \geq 0 \end{aligned} \quad (8-2)$$

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

**قضیه ۸-۱ (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  $b^T y \leq c^T x$  برقرار است.

درستی قضیه‌ی بالا به راحتی قابل تصدیق است زیرا  $b^T y \leq (Ax)^T y = x^T A^T y \leq x^T c = c^T x$ ، برقراری نامساوی‌ها از نامساوی‌های برنامه‌ی اولیه و دوگان حاصل می‌شود. قضیه‌ی قوی دوگانی در [۹] به صورت زیر بیان شده است.

<sup>۱۳</sup> Separation Oracle

**قضیه ۸-۲ (قضیه 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 برنامه‌ی اولیه و جواب  $y^*$  برای برنامه‌ی دوگان وجود خواهد داشت که  $c^T x^* = b^T y^*$ .

در صورتی مقادیر متغیرها محدود به اعداد صحیح شود به عنوان مثال  $x \in \{0, 1\}^n$  به این شکل از برنامه‌ریزی، برنامه‌ریزی صحیح می‌گوییم. این شکل از برنامه‌ریزی به سادگی قابل بهینه‌سازی نیستند. برداشتن محدودیت صحیح بودن متغیرها، برنامه‌ریزی خطی تعدیل شده 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 اولیه-دوگان [۹]) حاصل شده است. دقت کنید که جواب برنامه‌ریزی خطی تعدیل شده 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<sup>۱۴</sup> 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 ان‌پی-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.

---

<sup>۱۴</sup>Integrality Gap

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

جدول ۸-۱: 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<sup>۱۵</sup> 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  $ان‌پی$ -بهینه‌سازی (NP-Optimization) است،

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

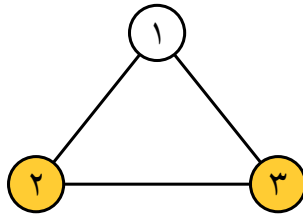
**تعریف ۸-۱** یک الگوریتم تقریبی برای مسئله‌ی  $P$  دارای ضریب تقریب  $\alpha$  است اگر برای هر نمونه‌ی  $I$  از  $P$ :

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

یک الگوریتم تقریبی با ضریب تقریب  $\alpha$ ، یک الگوریتم  $\alpha$ -تقریبی نامیده می‌شود. نمونه‌هایی از ضرایب تقریب متداول برای مسائل بهینه‌سازی در جدول ۸-۱ آمده است.

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

<sup>۱۵</sup>feasible



شکل ۸-۱: گراف  $G$  و یک پوشش رأسی برای آن

مسئله‌ی ۸-۲ (پوشش رأسی) گراف  $G = (V, E)$  و تابع هزینه‌ی  $w : V \rightarrow \mathbb{R}^+$  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 ۸-۱ 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 ۳ 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 ۸-۳  $\text{OPT} \leq |C| \leq 2 \text{OPT}$ .

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

$$|M| \leq \text{OPT}.$$

از طرفی می‌دانیم  $|C| = 2|M|$ . در نتیجه

$$|C| = 2|M| \leq 2 \text{OPT}.$$

□

بنا بر قضیه ۳-۸، الگوریتم ۳ یک الگوریتم ۲-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  $2n$  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.



# واژه‌نامه

## الف

pallet .....	پالت	heuristic .....	ابتکاری
robustness .....	پایداری	worth .....	ارزش
support .....	پشتیان	satisfiability .....	ارضاپذیری
convex hull .....	پوسته‌ی محدب	strategy .....	استراتژی
upper envelope .....	پوش بالایی	coalition .....	ائتلاف
covering .....	پوششی		

## ب

projective transformation .....	تبدیل تصویری	loading .....	بارگذاری
equilibrium .....	تعادل	game .....	بازی
relaxation .....	تعدیل	label .....	برچسب
intersection .....	تقاطع	linear programming .....	برنامه‌ریزی خطی
partition .....	تقسیم‌بندی	integer programming .....	برنامه‌ریزی صحیح
evolutionary .....	تکاملی	packing .....	بسته‌بندی
distributed .....	توزیع‌شده	best response .....	بهترین پاسخ
		maximum .....	بیشینه

## ج

brute-force .....	جست‌وجوی جامع
Depth-First Search .....	جست‌وجوی عمق‌اول

## پ

س	bin ..... جعبه
constructive ..... ساختی	
pay off, utility ..... سود	چ
	sink ..... چاله
ش	
quasi-polynomial ..... شبه‌چندجمله‌ای	ح
quasi-concave ..... شبه‌مقعر	action ..... حرکت
ص	خ
formal ..... صوری	selfish ..... خودخواهانه
	clique ..... خوشه
ع	د
rational ..... عاقل	binary ..... دودویی
agent-based ..... عامل-محور	dual ..... دوگان
action ..... عمل	bimatrix ..... دو ماتریسی
غ	ر
missing ..... غائب	vertex ..... رأس
decentralized ..... غیرمتمرکز	behaviour ..... رفتار
degenerate ..... غیرمعمول	coloring ..... رنگ‌آمیزی
ق	ز
transferable ..... قابل انتقال	scheduling ..... زمان‌بندی
lexicographically ..... قاموسی	biology ..... زیست‌شناسی
strong ..... قوی	

art gallery ..... نگارخانه‌ی هنر

gaurd ..... نگهبان

profile ..... نمایه

round-robin ..... نوبتی

## ک

minimum ..... کمینه

## م

subset sum ..... مجموع زیرمجموعه‌ها

set ..... مجموعه

pivot ..... محور

mixed ..... مختلط

hidden ..... مخفی

affine ..... مستوی

planar ..... مسطح

reasonable ..... منطقی

parallel ..... موازی

## و

facet ..... وجه

## ه

price of anarchy (POA) ..... هزینه‌ی آشوب

social cost ..... هزینه‌ی اجتماعی

price of stability (POS) ..... هزینه‌ی پایداری

## ی

edge ..... یال

isomorphism ..... یکرختی

## ن

outcome ..... نتیجه‌ی نهایی

Nash ..... نش

fixed point ..... نقطه ثابت

## **Abstract**

We present a standard template for typesetting theses in Persian. The template is based on the X<sub>Y</sub>TeX Persian package for the L<sup>A</sup>T<sub>E</sub>X typesetting system. This write-up shows a sample usage of this template.

**Keywords:** Thesis, Typesetting, Template, X<sub>Y</sub>TeX Persian



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