



دانشگاه صنعتی شریف
دانشکده‌ی مهندسی کامپیوتر

پروژه درس طراحی شی‌گرای سیستم‌ها

عنوان:

سامانه مدیریت حمل و نقل بار و کالا

نگارش:

مصطفی قدیمی

امیرحسن قتحی

استاد راهنما:

دکتر رامان رامسین

نیم‌سال دوم تحصیلی ۹۹-۹۸

سلام

فهرست مطالب

۸	۱ مروری بر نمودار UML
۸	۱-۱ مقدمه
۸	۲-۱ تعریف
۹	۳-۱ شکل‌گیری UML
۹	۴-۱ انواع نمودارها
۱۰	۵-۱ نسخه‌های UML
۱۰	۶-۱ نمودار کلاس
۱۱	۷-۱ نمودار Object
۱۳	۲ نحوه‌ی نگارش
۱۳	۱-۲ پرونده‌ها
۱۳	۲-۲ عبارات ریاضی
۱۴	۳-۲ علائم ریاضی پرکاربرد
۱۵	۴-۲ لیست‌ها
۱۵	۵-۲ درج شکل
۱۶	۶-۲ درج جدول

۱۶	۷-۲ درج الگوریتم
۱۷	۸-۲ محیط‌های ویژه
۱۸	۳ برخی نکات نگارشی
۱۸	۱-۳ فاصله‌گذاری
۱۹	۲-۳ شکل حروف
۱۹	۳-۳ جدانویسی
۲۰	۴-۳ جدانویسی مرجح
۲۱	۴ مقدمه
۲۱	۱-۴ تعریف مسئله
۲۲	۲-۴ اهمیت موضوع
۲۲	۳-۴ ادبیات موضوع
۲۴	۴-۴ اهداف تحقیق
۲۴	۵-۴ ساختار پایان‌نامه
۲۵	۵ مفاهیم اولیه
۲۵	۱-۵ برنامه‌ریزی خطی
۲۷	۲-۵ الگوریتم‌های تقریبی
۲۸	۳-۵ پوشش رأسی
۳۱	۶ کارهای پیشین
۳۲	آ مطالب تکمیلی

فهرست شکل‌ها

۱۵	۱-۲ یک گراف و پوشش رأسی آن
۱۵	۲-۲ یک گراف جهت‌دار بدون دور
۲۹	۱-۵ گراف G و یک پوشش رأسی برای آن

فهرست جدول‌ها

۱۶ ۱-۲ عملگرهای مقایسه‌ای
۲۸ ۱-۵ نمونه‌هایی از ضرایب تقریب برای مسائل بهینه‌سازی

فصل ۱

مروری بر نمودار UML

۱-۱ مقدمه

مدر این فصل به مرور و آشنایی با نمودارهای زبان مدل سازی یک پارچه خواهیم پرداخت.

۲-۱ تعریف

UML یا زبان مدل سازی یک پارچه^۱، زبان استاندارد مهندسی نرم افزار برای مدل سازی است. مجموعه ای از بهترین رویه های عملی^۲ را برای مدل کردن سیستم های بزرگ و پیچیده، ارائه می دهد.

کاربرد: برای مشخص کردن^۳، مجسم کردن^۴، ساختن^۵ و مستند کردن^۶ محصولات^۷ مهندسی نرم افزار و همچنین مدل سازی کسب و کار از آن استفاده می شود.

سوال: در چه صورت می توان از UML استفاده کرد؟ در صورتی که بتوان برای برای آن سیستم object متصور د و رفتار سیستم را بتوان به صورت تعامل مجموعه ای از اشیاء نمایش داد.

^۱ Unified Modeling Language

^۲ practice

^۳ specifying

^۴ visualizing

^۵ constructing

^۶ documenting

^۷ artifact در این جا به معنی مصنوع و محصول است.

۳-۱ شکل‌گیری UML

این اسلاید، تاثیر زبان‌های مدل‌سازی (بعضی از مستطیل‌های طوسی مانند C++) را بر روی UML نمایش می‌دهد.

نکته ۱: روش CRC^۸ یکی از مهم‌ترین روش‌ها برای تشخیص کلاس است.

نکته ۲: منظور از Formal Specification تعریف‌های ریاضی است.

نکته ۳: تاثیرگذارترین متدولوژی‌ها روی شکل‌گیری UML عبارتند از:

۱. Objectory

۲. Booch

۳. OMT

زیرا پایه‌گذاران این متدولوژی‌ها، UML را نیز ابداع کرده‌اند.

۴-۱ انواع نمودارها

در مهندسی نرم‌افزار انواع نمودارها به سه دسته‌ی زیر تقسیم می‌شوند:

۱. **ساختاری یا Structural**: سیستم از چه اجزایی تشکیل شده است و چه رابطه‌ای با یکدیگر دارند؛ مانند نمودار کلاس.

۲. **رفتاری یا Behavioral**: اجزای داخلی چگونه با یکدیگر کار می‌کنند و ترتیب انجام کارها به چه شکلی است به عبارت دیگر تقدم-تاخر وجود دارد؛ مانند نمودار flowchart

۳. **وظیفه‌ای یا Functional**: UML این نمودار را ندارد. کارها را بدون ترتیب توصیف می‌کند. مثل این‌که سقف را از روی یک سازمان برداریم، کارمندان و ارتباطاتشان را مشاهده کنیم.

نمودار بسته: از این نمودار فقط در جریان کاری تحلیل استفاده می‌شود. وظیفه‌اش افزاز کلاس‌ها به تعدادی بسته است.

نمودار مورد کاربرد: در این نمودار مشخص می‌شود که کنش‌گرهای سیستم چه سرویس‌هایی از آن می‌گیرند. هر مورد کاربرد یک functionality است. علاوه بر این رابطه‌ی بین موارد کاربرد را مشاهده می‌کنیم و ترتیب ندارند. پس به همین دلیل جزو نمودارهای رفتاری یا Behavioral نیست، اما همان‌طور که گفته شد، نمودار UML، نمودارهای وظیفه‌ای را ندارد و به همین علت در آن، جزو نمودارهای وظیفه‌ای محسوب می‌شود.

نکته: به نمودار فعالیت، فلوچارت شی‌گرا می‌گویند.

نمودار زمانی: زمانی که محدودیت زمان پاسخ داریم از این نمودار استفاده می‌کنیم؛ مانند سیستم‌های بی‌درنگ.

۵-۱ نسخه‌های UML

در این قسمت تکامل نسخه‌های مختلف UML و مفاهیم جدید آن مورد بررسی قرار گرفته شده است.

۶-۱ نمودار کلاس

هر کلاس یک مستطیل چند قسمتی^۹ است که عموماً سه قسمت است. در جریان کاری تحلیل، وارد جزئیات کلاس‌ها نمی‌شویم؛ مثلاً نوع attribute یا access visibility مشخص نمی‌شود.

نکته: هر توارشی GenSpec نیست اما هر GenSpec، یک رابطه‌ی توارث است.

Refused Bequest: یک ویژگی از پدر به فرزند رسیده است اما فرزند به آن نیازی ندارد. در این حالت چون رابطه‌ی IS-A نقض می‌شود، می‌توانیم بگوییم که GenSpec نیست. مثلاً برای کلاس اسب، متد خوردن و راه رفتن داریم. برای کلاس عنکبوت هم همین دو متد را نیز داریم. اگر عنکبوت را زیرکلاس اسب در نظر بگیریم، یکی از فاجعه‌بارترین اشتباهات است؛ زیرا از inheritance برای reuse استفاده کرده‌ایم. اگر چهار نعل و یورتمه را برای اسب، تعریف کنیم، حالت Refused Bequest به وجود می‌آید که مشخص می‌کند رابطه دیگر رابطه‌ی IS-A نیست.

رابطه‌ی Association: یک رابطه‌ی مانا بین کلاس‌ها برقرار است. در پیاده‌سازی معمولاً برای این رابطه یک attribute مانند Foreign Key تعریف می‌کنیم. اگر این رابطه جهت‌دار باشد، یعنی دید از یک طرف به سمت دیگر وجود ندارد. (مثالی که تو جزوه نوشتی بیارم)

رابطه‌ی Aggregation: یک رابطه‌ی Association، Aggregation است، اگر یکی از سه شرط زیر را داشته

^۹ multi compartment

باشد (در این رابطه معمولاً دید یک طرفه از کل به جزء وجود دارد):

۱. تشکیل شدن یا Assembly: وقتی یک طرف از instance های طرف دیگر تشکیل شده است؛ مانند درخت و برگ‌هایش.

۲. محتوی بودن یا Containment: وقتی یک طرف Container طرف دیگر بشود؛ مانند بسته‌ی پستی و محتویات داخل آن.

۳. عضویت یا Membership: وقتی یک طرف عضو طرف دیگر باشد؛ مانند یک تیم فوتبال و اعضای آن.

رابطه‌ی Composition:

یک رابطه‌ی Composition، Aggregation است، اگر:

۱. Lifetime dependency داشته باشیم؛ یعنی با از بین رفتن کل، جزء هم از بین برود.

۲. sharing نداشته باشیم.

نکته: با توجه به نکات گفته شده، رابطه‌ی Assembly به احتمال زیاد Composition است؛ زیرا Lifetime dependency دارد.

رابطه Dependency: تغییرات در B موجب تغییرات در A می‌شود؛ برای مثال اگر به کلاس A پیغام ارسال کنیم و به عنوان پارامتر، نمونه‌ی کلاس B را بفرستیم، یک دید موقت به وجود می‌آید. برقراری دید از طریق پارامتر معمولاً غیرمانا است.

بسیاری از مدل‌سازها در ابتدای کار، همه‌ی روابط را dependency در نظر می‌گیرند و سپس به روابط دیگر تبدیل می‌کنند:

$$Dependency \rightarrow Association \rightarrow Aggregation \rightarrow Composition$$

$$Dependency \rightarrow GenSpec$$

۷-۱ نمودار Object

این نمودار ساختار Object ها را در زمان اجرا نشان می‌دهد.

موارد کاربرد: (۱) در بعضی از مواقع نمودار کلاس ایجاد ابهام می‌کند. مثلاً نمودار کلاس دیارتمان به خودش رابطه‌ی زیردیارتمان دارد که در این اسلاید نمایش داده شده است. (۲) برای بیان وضعیت استثنایی که با نمودار کلاس قابل بیان نیست؛ برای مثال در فوتبال کلاس زمین، بازیکن و ... داریم اما برای بیان وضعیت آفساید به نمودار Object نیاز داریم.

فصل ۲

نحوه‌ی نگارش

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

۱-۲ 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.

۲-۲ 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.

۲-۳ 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: $\{1, 2, 3\}$
- 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: $\infty, \emptyset, \forall, \exists, \Delta, \angle, \ell, \equiv, \dots$

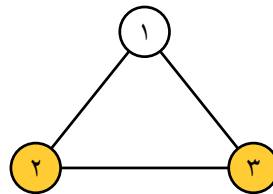
۲-۴ لیست‌ها

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

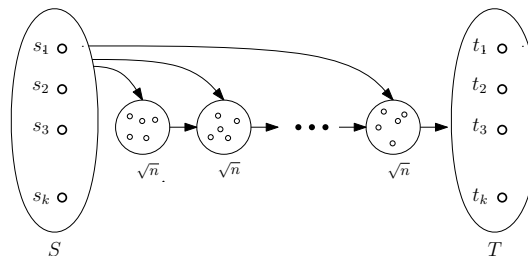
۲-۵ درج شکل

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 ۲-۱ نمونه‌ای از اشکال ایجادشده با این ابزار را نشان می‌دهد.



شکل ۲-۱: 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 ۲-۲ را ببینید.



شکل ۲-۲: یک گراف جهت‌دار بدون دور

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

جدول ۱-۲: عملگرهای مقایسه‌ای

۶-۲ 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 ۱-۲ را ببینید.

۷-۲ درج الگوریتم

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

الگوریتم ۱ 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

۸-۲ 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 ۱-۲ (اصل لانه‌کبوتری) اگر $n + 1$ یا بیشتر کبوتر درون n لانه قرار گیرند، آنگاه لانه‌ای وجود دارد که شامل حداقل دو کبوتر است.

فصل ۳

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.

۳-۱ 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: کلاس 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.

۲-۳ 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 صفحه‌کلید فارسی استاندارد^۲ 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 اضافه‌ی 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.

۳-۳ 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 «به» صفت یا فعل ساخته است. مانند: بسزا، ببینم.
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.

^۲ صفحه‌کلید فارسی استاندارد برای ویندوز، تهیه‌شده توسط بهنام اسفهبید

۳-۴ جدانویسی مرجع

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

تبصره: اجزای منتهی به «هـاء 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.

فصل ۴

مقدمه

نخستین فصل یک پایان نامه به معرفی مسئله، بیان اهمیت موضوع، ادبیات موضوع، اهداف تحقیق و معرفی ساختار پایان نامه می پردازد. در این فصل نمونه ای از این مقدمه آورده شده است.^۱

۴-۱ تعریف مسئله

مسئله ی مسیریابی وسایل نقلیه حالت کلی تر مسئله ی فروشنده ی دوره گرد^۲ و یکی از مسائل جالب در حوزه ی بهینه سازی ترکیباتی است. در این مسئله، تعدادی وسیله ی نقلیه که هر کدام در انبار^۳ مشخصی قرار دارند به همراه تعدادی مشتری در قالب یک گراف داده شده است که گره های این گراف نشان دهنده ی مشتریان و انبارها است و وزن یال های گراف نشان دهنده ی هزینه ی حرکت بین گره های مختلف می باشد. هدف، یافتن دورهای مجزایی برای هر وسیله می باشد به نحوی که این دورها در برگیرنده ی تمام مشتریان بوده و مجموع هزینه ی دورها کمینه گردد.

گونه های مختلفی از مسئله ی مسیریابی وسایل نقلیه با محدودیت های متفاوت توسط پژوهشگران مورد مطالعه قرار گرفته است. از جمله در نظر گرفتن محدودیت هایی نظیر پنجره ی زمانی، به این مفهوم که هر مشتری در بازه ی زمانی خاصی باید ملاقات شود و یا در نظر گرفتن محدودیت برای ظرفیت وسایل که سبب می شود هر وسیله تنها تا زمانی بتواند به مشتریانی سرویس دهی کند که سطح تقاضای آنها از ظرفیت وسیله تجاوز نکند.

از جمله گونه هایی که اخیراً مورد توجه قرار گرفته، و تا حد زیادی به مسائل دنیای واقعی شبیه تر است، مسئله ی مسیریابی

^۱ مطالب این فصل نمونه از پایان نامه ی آقای حسام الدین منفرد گرفته شده است.

^۲ Travelling Salesman Problem

^۳ Depot

وسایل نقلیه‌ی ناهمگن^۴ می‌باشد. در این گونه از مسئله، وسایل نقلیه ناهمگن در نظر گرفته می‌شوند، به این معنی که هزینه‌ی پیمایش یال‌ها برای هر وسیله‌ی نقلیه می‌تواند متفاوت باشد. تعریف دقیق‌تر این مسئله در زیر آمده است.

مسئله‌ی ۱-۴ 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.

۲-۴ اهمیت موضوع

مسئله‌ی مسیریابی وسایل نقلیه کاربردهای بسیار گسترده‌ای در حوزه‌ی حمل و نقل دارد. برای نخستین بار این مسئله برای مسیریابی تانکرهای سوخت‌رسان مطرح شد [۹]. 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 [۹، ۱۰، ۱۱] بیان شده است.

۳-۴ ادبیات موضوع

همان‌طور که ذکر شد مسئله‌ی مسیریابی وسایل نقلیه‌ی ناهمگن صورت عمومی مسئله‌ی فروشنده دوره‌گرد می‌باشد. مسئله‌ی فروشنده‌ی دوره‌گرد در حوزه‌ی مسائل ان‌پی-سخت^۵ 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^۶ استفاده می‌شود.

Heterogeneous Vehicle Routing Problem^۴

NP-hard^۵

Approximation Algorithm^۶

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^۷ و میانبر کردن^۸ دورهای بدست آمده، الگوریتمی با ضریب تقریب ۲ ارائه می‌شود. در الگوریتم دوم که متعلق به کریستوفایدز^۹ [۹] است، به کمک ساخت دور اویلری^{۱۰} بر روی اجتماع یال‌های درخت پوشای کمینه و یال‌های تطابق کامل کمینه^{۱۱} 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 [۹]. در روش مطرح شده بعد از 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 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^{۱۲} و ساخت درخت پوشای کمینه [۹]، ضریب تقریب $1/5m$ به کمک حل تعدیل برنامه‌ریزی خطی با روش بیضی^{۱۳} و اعمال الگوریتم کریستوفایدز [۹] و ضریب تقریب ۲ به کمک راه حل اولیه-دوگان^{۱۴} 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 [۹].

Minimum Spanning Tree^۷

Shortcut^۸

Christofides^۹

Eulerian Cycle^{۱۰}

Minimum Perfect Matching^{۱۱}

Linear Programming Relaxation^{۱۲}

Ellipsoid Method^{۱۳}

Primal-Dual^{۱۴}

۴-۴ اهداف تحقیق

در این پایان نامه سعی می شود که مسئله ی مسیریابی وسایل نقلیه برای زیرگراف های ناهمگن مورد مطالعه قرار گیرد. از جمله زیرگراف های مورد نظر ما دور، درخت و مسیر می باشد. بعد از مطالعه ی کارهای انجام شده در این زمینه سعی می شود که مسئله به صورت دقیق تر مورد بررسی قرار گیرد.

۴-۵ ساختار پایان نامه

این پایان نامه شامل پنج فصل است. فصل دوم دربرگیرنده ی تعاریف اولیه ی مرتبط با پایان نامه است. در فصل سوم مسئله ی 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، مسئله ی درخت های ناهمگن در چهار شکل مختلف مورد بررسی قرار می گیرد. سپس نگاهی کوتاه به مسئله ی مسیرهای ناهمگن خواهیم داشت. در انتها با تغییر تابع هدف، به حل مسئله ی کمینه کردن حداکثر اندازه ی درخت ها می پردازیم. فصل پنجم به نتیجه گیری و پیش نهادهایی برای کارهای آتی خواهد پرداخت.

فصل ۵

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.

۵-۱ 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.

$$\text{minimize } c^T x \quad (5-1)$$

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

$$x \geq 0$$

در روابط فوق، x بردار متغیرها، b, c بردارهای ثابت و A ماتریس ضرایب می‌باشد. به سادگی قابل مشاهده است که رابطه‌ی (۵-۱) 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 (۵-۱) یا اضافه کردن پارامتر و متغیر جدید قابل مدل کردن می‌باشد. 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 بیضی [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^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 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 (۵-۱) به صورت زیر می‌باشد:

$$\text{minimize } b^T y \quad (5-2)$$

$$\text{s.t. } A^T y \leq c$$

$$y \geq 0$$

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

قضیه‌ی ۵-۱ (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 $c^T x = x^T A^T y \leq x^T c = c^T x$ ، برقراری نامساوی‌ها از نامساوی‌های برنامه‌ی اولیه و دوگان حاصل می‌شود. قضیه‌ی قوی دوگانی در [۴] به صورت زیر بیان شده است.

قضیه‌ی ۵-۲ (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 برنامه‌ی اولیه و جواب y^* برای برنامه‌ی دوگان وجود خواهد داشت که $c^T x^* = b^T y^*$.

^۱Separation Oracle

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 [۴]) 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^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.

۵-۲ 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.

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 ان‌پی-بهینه‌سازی (NP-Optimization) است،

^۲Integrality Gap
^۳feasible

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

جدول ۵-۱: نمونه‌هایی از ضرایب تقریب برای مسائل بهینه‌سازی

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

تعریف ۵-۱ یک الگوریتم تقریبی برای مسئله‌ی P دارای ضریب تقریب α است اگر برای هر نمونه‌ی I از P :

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

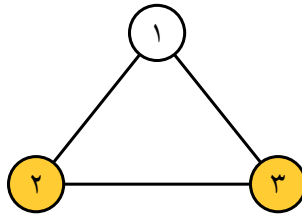
یک الگوریتم تقریبی با ضریب تقریب α ، یک الگوریتم α -تقریبی نامیده می‌شود. نمونه‌هایی از ضرایب تقریب متداول برای مسائل بهینه‌سازی در جدول ۵-۱ آمده است.

۵-۳ پوشش رأسی

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

مسئله‌ی ۵-۱ (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 ۵-۱ نمونه‌ای از یک پوشش رأسی را نشان می‌دهد. در زیر یک الگوریتم حریصانه برای مسئله‌ی پوشش رأسی غیروزن‌دار ارائه شده است.



شکل ۵-۱: گراف G و یک پوشش رأسی برای آن

الگوریتم ۲ 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 ۲ 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 ۵-۳ $\text{OPT} \leq |C| \leq 2 \text{OPT}$.

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 ۳-۵، الگوریتم ۲ 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 ۶

کارهای پیشین

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

پیوست آ

مطالب تکمیلی

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

واژه‌نامه

covering پوششی

الف

heuristic ابتکاری

ت

projective transformation تبدیل تصویری

ب

equilibrium تعادل

loading بارگذاری

relaxation تعدیل

game بازی

intersection تقاطع

label برچسب

partition تقسیم‌بندی

linear programming برنامه‌ریزی خطی

evolutionary تکاملی

integer programming برنامه‌ریزی صحیح

distributed توزیع‌شده

packing بسته‌بندی

best response بهترین پاسخ

maximum بیشینه

ج

brute-force جست‌وجوی جامع

پ

چ

sink چاله

pallet پالت

robustness پایداری

support پشتیبان

ح

action حرکت

convex hull پوسته‌ی محدب

upper envelope پوش بالایی

غ

missing غائب

خ

selfish خودخواهانه

ق

transferable قابل انتقال

د

binary دودویی

ک

minimum کمینه

ر

practice رویه عملی

م

deployment مستقرسازی

ز

artifact مصنوع، محصول
scheduling زمان‌بندی

ن

entity-relational نهاد رابطه
constructive ساختی

س

و

facet وجه
quasi-polynomial شبه‌چندجمله‌ای

ش

ه

price of anarchy (POA) هزینه‌ی آشوب
formal صوری

ص

ی

edge یال
rational عاقل

ع