دانشکده علوم

گروه ریاضی

پایان نامه کارشناسی ارشد

عنوان:

الگوریتم‌های تقریبی برای مسأله‌ی مکان‌یابی تسهیلات بدون ظرفیت

نگارنده:

سحر صلواتی

استاد راهنما:

دکتر محمد حسینی‌کولایی

مهر ۱۳۹۰

إنه من سليمان وإنه

بسم الله الرحمن الرحيم

تقدیم به

پدر و مادر عزیزم

به پاس وجودشان

کاشکی هســتی زبانی داشتی تا ز هستان پـرده‌ها بـرداشــتی
هرچه گویی ای دم هستی از آن پرده‌ای دیگر بر او بستی بدان

مولانا

سپاس خداوند یکتا و آفریدگار توانا را که همه‌ی خوبی‌ها از اوست و بزرگی سزاوار او.

مراتب سپاس و قدردانی خود را از جناب آقای دکتر محمد حسینی‌کولایی، استاد محترم راهنمای پایان‌نامه که با راهنمایی‌های دقیق و مستمر خود در شکل‌گیری این پایان‌نامه نقش مهمی داشته‌اند، ابراز می‌کنم و شخصیت علمی و اخلاقی ایشان را که برای اینجانب الگویی پسندیده بوده‌اند، تحسین می‌نمایم.

همچنین از جناب آقای دکتر علی باستانی و جناب آقای دکتر هادی خطیب‌زاده که زحمت داوری این پایان‌نامه را پذیرفتند، سپاسگزاری می‌نمایم.

از دوست عزیزم خانم نیره ذوالفقاری که همچون خواهری دلسوز در کنارم بود سپاسگزارم.

در پایان از خانواده عزیزم به پاس عاطفه‌ی سرشار و گرمای امید بخش وجودشان که در سردترین روزگاران، بهترین پشتیبان من بودند، نهایت تشکر را دارم.

چکیده

مسأله‌ی مکان‌یابی تسهیلات بدون ظرفیت، یک مسأله از دسته مسائل $\mathcal{NP}$–کامل می‌باشد. از آنجا که الگوریتم زمان چند جمله‌ای برای حل این نوع مسائل وجود ندارد، الگوریتم‌های تقریبی با زمان اجرای چند جمله‌ای برای یافتن جوابی نزدیک به جواب بهینه مورد استفاده قرار می‌گیرند. در این پایان نامه بررسی و تحلیل الگوریتم‌های تقریبی مد نظر است که از برنامه‌ریزی خطی برای طراحی آنها استفاده شده است. در اینجا از روش متناسب کردن دوگان برای تحلیل الگوریتم‌ها و بررسی نسبت تقریب آنها استفاده می‌کنیم و با استفاده از یک مسأله‌ی برنامه‌ریزی خطی آشکار ساز نسبت، مقدار نسبت تقریب الگوریتم‌ها محاسبه می‌شود. بهترین الگوریتمی که بررسی می‌شود به نسبت تقریب ۱/۵۲ می‌رسد و زمان اجرای شبه خطی دارد.

واژگان کلیدی: الگوریتم‌های تقریبی، مسأله‌ی مکان‌یابی تسهیلات بدون ظرفیت، روش حریصانه، برنامه‌ریزی خطی آزاد شده، برنامه‌ریزی خطی صحیح.

فهرست مندرجات

مقدمه

یکی از ابعاد مهم فرآیندهای کاری سازمان‌ها و صنایع که در بهینه‌سازی مورد توجه قرار می‌گیرد، صرفه اقتصادی و دسترسی کارای گروهی خدمت گیرنده در دست‌یابی به مجموعه‌ای از زیر ساخت‌ها یا تسهیلات است. به عنوان نمونه می‌توان به پیاده‌سازی یک زنجیره‌ی تأمین از یک کسب و کار مانند مکان‌یابی خدمات ضروری، همچون مراقبت‌های پزشکی، آموزش و یا ایجاد شبکه‌های حمل و نقل اشاره کرد. مسأله‌ی استاندارد مکان‌یابی تسهیلات به عنوان یک مسأله‌ی رایج در مبحث تحقیق در عملیات به ارائه‌ی یک فرمول ریاضی برای جنبه‌های بهینه‌سازی این چنین کاربردهایی می‌پردازد.

در ادامه به بیان مثالی از مسأله‌ی مکان‌یابی تسهیلات در دنیای واقعی می‌پردازیم. فرض کنید یک مرکز تهیه و توزیع غذا در حال برنامه‌ریزی برای تأسیس شعبه‌هایی است تا بتواند راحت تر و با هزینه‌ی کمتر به مشتریان خدمات دهد. این مرکز، چندین محل را برای تأسیس شعبه‌های خود در نظر دارد. هزینه‌ی تأسیس هر شعبه با توجه به محل آن برای مرکز مورد نظر مشخص است. همچنین موقعیت مکانی مشتری‌ها که سفارش غذا خواهند داد نیز معلوم است. پرسش این است که مرکز باید چه تعداد از این شعبه‌ها و در چه محل‌هایی تأسیس کند تا تأمین خواسته‌های مشتریان، با کمترین هزینه از حیث تأسیس شعبه‌ها و همچنین فاصله مشتریان از شعبه‌های مذکور انجام پذیرد.

اساساً یک مسأله‌ی استاندارد مکان‌یابی تسهیلات از طریق چهار شاخص زیر شناسایی می‌شود:

۱) یک مجموعه از نقاط به طوری که تسهیلات در آن نقاط باید دایر شوند یا مورد بهره برداری قرار گیرند. هزینه‌ی برپایی برای هر نقطه مشخص است.

۲) یک مجموعه از نقاط تقاضا که مشتریان در آنها قرار می‌گیرند. هر یک از این نقاط تقاضا باید به منظور تأمین نیازش به یکی از تسهیلات دایر متصل شود تا بتواند از آن خدمات بگیرد. این تخصیص باید به طریقی صورت گیرد که در نهایت هزینه‌ی کل مینیمم گردد. هزینه‌ی کل، مجموع هزینه‌ی اتصال نقاط تقاضا یا مشتریان به تسهیلات و هزینه‌ی برپایی تسهیلات است.

۳) یک گروه از تقاضاها که باید توسط تسهیلات تأمین شوند.

۴) یک تابع هدف که اگر مجموعه‌ای مشخص از تسهیلات دایر شوند تا مجموعه‌ای از سفارشات مشتریان تأمین شود، مقدار آن با توجه به این دو مجموعه مینیمم گردد.

درحالت کلی، مسألهٔ مکان‌یابی تسهیلات بدون ظرفیت [۱] یا به اختصار $UFLP$، که پایه‌ای‌ترین مسألهٔ مکان‌یابی تسهیلات است، شامل مجموعهٔ $\mathcal{F}$ با تعداد n_f تسهیلات، مجموعهٔ $\mathcal{C}$ با تعداد n_c شهر، یک هزینهٔ f_i برای باز کردن تسهیلات $i \in \mathcal{F}$ و یک هزینهٔ اتصال c_{ij} برای اتصال شهر j به تسهیلات i می‌باشد. هدف، باز کردن زیر مجموعه‌ای از تسهیلات در $\mathcal{F}$ و اتصال هر شهر به یک تسهیلات باز، است به طوری که هزینهٔ کل که مجموع هزینهٔ تسهیلات باز شده و هزینهٔ اتصال شهرها به تسهیلات باز است، مینیمم گردد. در اینجا هزینه‌های اتصال را متریک در نظر می‌گیریم یعنی در نامساوی مثلثی صدق می‌کنند.

این گروه مسائل از دسته مسائل $\mathcal{NP}$–کامل هستند، یعنی برای حل آنها الگوریتمی که در زمان چندجمله‌ای مسأله را حل کند وجود ندارد. بنابراین الگوریتم‌هایی که راه حل‌های نزدیک به راه حل بهینه را محاسبه می‌کنند مورد توجه قرار گرفتند. روش‌های حل مختلفی برای این نوع مسائل وجود دارد که از آن جمله می‌توان به روش‌های مکاشفه‌ای [۲]، جستجوی محلی [۳] و الگوریتم‌های تقریبی [۴] مانند الگوریتم‌های گرد کردن LP [۵] و اولیه–دوگان [۶] اشاره کرد.

در سالهای اخیر، الگوریتم‌های تقریبی برای حل مسائل مرتبط با مکان‌یابی تسهیلات، توجه بسیاری از محققین را به خود جلب کرده است. اولین الگوریتم تقریبی با فاکتور تقریبی ثابت توسط شمویز [۷]، تردوس [۸] و آردل [۹] [۱۶]، ارائه شد که مقدار آن ۳/۱۶ بود. الگوریتم‌های تقریبی بسیاری برای این مسأله پیشنهاد شده است [۱, ۳, ۴, ۵, ۸, ۱۰, ۱۲, ۱۷, ۱۹]. جدول ۱ خلاصه‌ای از این نتایج را نشان می‌دهد.

[۱] Uncapacitated Facility Location Problem
[۲] Heuristic
[۳] Local search
[۴] Approximation Algorithms
[۵] Rounding
[۶] Primal-Dual
[۷] Shmoys
[۸] Tardos
[۹] Aardal

روش مورد استفاده / زمان اجرا	مرجع	نسبت تقریب
الگوریتم حریصانه / $O(n^3)$	هوچبااوم [۷]	$\ln n_c$
گرد کردن LP	شمویز [۱۶]	۳/۱۶
فزایش حریصانه + گرد کردن LP	گیوها و کالر [۵]	۲/۴۱
گرد کردن LP	چوداک و شمویز [۴]	۱/۷۳۶
جستجوی محلی / $O(n^6 \log(n/\epsilon))$	کروپولو [۱۲]	$5 + \epsilon$
روش اولیه—دوگان / $O(n^2 \log n)$	جاین و وزیرانی [۱۰]	۳
فزایش حریصانه + روش اولیه—دوگان / $O(n^3)$	چاریکار و گیوها [۳]	۱/۸۵۳
فزایش حریصانه + روش اولیه—دوگان + رند کردن LP	چاریکار و گیوها [۳]	۱/۷۲۸
الگوریتم حریصانه / $O(m \log m)$	مهدیان، جاین، صابری و مارکاکیس [۱۳,۸]	۱/۸۶۱*
الگوریتم حریصانه / $O(n^3)$	جاین، مهدیان و صابری [۹,۸]	۱/۶۱*
گرد کردن LP	سویریدنکو [۱۷]	۱/۵۸۲
مقیاس‌بندی هزینه + الگوریتم حریصانه / $\tilde{O}(n)$	مهدیان، یه، ژنگ [۱۴]	۱/۵۲*

جدول ۱: نتایج الگوریتم‌های تقریبی برای مسأله‌ی مکان‌یابی تسهیلات.
نسبت‌های تقریبی که با ستاره مشخص شده‌اند در این پایان نامه بررسی شده‌اند.

اگر بین تمام نقاط تسهیلات و شهرها یک گراف دو بخشی کامل در نظر بگیریم، آنگاه m و n به ترتیب تعداد کل یال‌ها و رأس‌های این گراف دو بخشی کامل هستند. به عبارت دیگر $m = n_c n_f$ و $n = n_c + n_f$.

گیوها[۱۰] و کالر[۱۱][۵]، نشان دادند که رسیدن به یک ضمانت تقریبی ۱/۴۶۳ غیر ممکن است مگر اینکه $\mathcal{NP} \subseteq DTIME[n^{O(\log \log n)}]$ باشد ($DTIME(t)$ دسته‌ای از مسائل می‌باشد که برای آنها یک الگوریتم قطعی[۱۲] با زمان اجرای $O(t)$ وجود دارد). این نتیجه توسط سویریدنکو اصلاح شد و او نشان داد که نسبت تقریب الگوریتم تقریبی برای $UFLP$ نمی تواند کمتر از ۱/۴۶۷ باشد مگر این که $\mathcal{P} = \mathcal{NP}$ [۱۷].

در این پایان نامه، ابتدا در فصل اول، به بیان مفاهیم پایه‌ای همچون پیچیدگی محاسباتی، برنامه‌ریزی خطی، برنامه‌ریزی صحیح و مفهوم الگوریتم تقریبی می‌پردازیم. درفصل دوم، به معرفی مسأله‌ی مکان‌یابی تسهیلات و چگونگی مدل‌بندی ریاضی آن پرداخته و سپس از طریق روش متناسب کردن دوگان[۱۳]، یک الگوریتم تقریبی برپایه‌ی برنامه‌ریزی خطی برای مسأله‌ی مکان‌یابی تسهیلات بدون ظرفیت، ارائه می‌دهیم و در ادامه با به دست

---

[۱۰] Guha

[۱۱] Khuller

[۱۲] Deterministic Algorithm

[۱۳] Dual Fitting

آوردن مسأله‌ی آشکار ساز نسبت[۱۴] و حل آن، مقدار نسبت تقریب الگوریتم را محاسبه می‌کنیم [۸, ۱۱, ۱۳, ۱۴]. فصل سوم، شامل الگوریتم حریصانه‌ی[۱۵] جاین[۱۶]، مهدیان[۱۷] و صابری[۱۸] [۸, ۹] خواهد بود که به یک نسبت تقریب بهتر برای مسأله می‌رسد. سرانجام در فصل چهارم، به کمک روش مقیاس‌بندی هزینه[۱۹]، الگوریتمی با بهترین نسبت تقریب شناخته شده برای مسأله‌ی مکان‌یابی تسهیلات، به دست می‌آوریم که می‌تواند در زمان شبه خطی[۲۰]، اجرا شود [۳, ۵, ۱۰, ۱۴, ۱۵].

---

[۱۴] Factor Revealing LP

[۱۵] Greedy Algorithm

[۱۶] Jain

[۱۷] Mahdian

[۱۸] Saberi

[۱۹] Cost Scaling

[۲۰] Quasi-Linear Time

فصل ۱

تعاریف و مفاهیم اولیه

در این فصل به بیان تعاریف و مفاهیم اولیه‌ی مورد نیاز در این پایان‌نامه می‌پردازیم.

۱.۱ برنامه‌ریزی خطی

برنامه‌ریزی خطی[۱] (LP) یک روش ریاضی برای بهینه‌سازی یک تابع هدف خطی (از قبیل ماکزیمم نمودن سود یا مینیمم کردن هزینه) تحت محدودیت‌های خطی مساوی و/ یا نامساوی است. برای نشان دادن یک مسأله‌ی بهینه‌سازی به صورت یک مدل برنامه‌ریزی خطی، چهار فرض الزامی است. این چهار فرض عبارت‌اند از تناسب، جمع‌پذیری، بخش‌پذیری و معین بودن.

مسائل LP را به طور عمده به دو شکل استاندارد و متعارفی نمایش می‌دهند.

یک مسأله‌ی برنامه‌ریزی خطی به فرم استاندارد به صورت

$$\begin{aligned} min \quad & c^T x \\ s.t. \quad & Ax = b, \\ & x \geq 0. \end{aligned} \tag{۱.۱}$$

[۱] Linear Programming

نمایش داده می‌شود که در آن c^Tx **تابع هدف**، $x \in \mathbb{R}^m$ بردار متغیرها، $c \in \mathbb{R}^m$ بردار ضرایب هزینه و $A \in \mathbb{R}^{n\times m}$ ماتریس ضرایب می‌باشند. معادلات $Ax = b$ نیز **محدودیت‌های خطی** نامیده می‌شوند.

شکل دیگر نمایش مسأله‌ی LP استفاده از شکل متعارفی است که به صورت زیر نمایش داده می‌شود:

$$\begin{aligned} min \quad & c^Tx \\ s.t. \quad & Ax \leq b, \\ & x \geq \circ. \end{aligned}$$

در عمل بسیاری از مسائل در حوزه‌ی تحقیق در عملیات را می‌توان در قالب مسائل LP مدل‌بندی کرد.

تعریف ۱.۱ در یک مسأله‌ی برنامه‌ریزی خطی (LP)، x را یک **جواب شدنی** گویند هرگاه در تمام محدودیت‌های مسأله صدق کند. مجموعه‌ی جواب‌های شدنی را نیز **مجموعه‌ی شدنی** یا **ناحیه شدنی** گویند که معمولاً با مجموعه‌ی X نمایش داده می‌شود. فرض کنید $A = (B, N)$ یک افراز از ماتریس ضرایب A باشد که در آن B یک ماتریس معکوس‌پذیر $n \times n$ و N یک ماتریس $n \times (m-n)$ باشد. $x = \begin{pmatrix} x_B \\ x_N \end{pmatrix}$ را که در آن $x_B = B^{-1}b$ و $x_N = \circ$ یک **جواب پایه‌ای** و اگر $x_B \geq \circ$، آن را یک **جواب شدنی پایه‌ای**[۱] (BFS) می‌نامند. اگر $x_B > \circ$، آنگاه x_B یک **جواب شدنی پایه‌ای ناتباهیده** است و در صورتی که یکی از مؤلفه‌های x_B صفر باشد، آنگاه x_B یک **جواب شدنی پایه‌ای تباهیده** نامیده می‌شود. همچنین، مؤلفه‌های x_B به عنوان **متغیرهای پایه‌ای** و مؤلفه‌های x_N به عنوان **متغیرهای غیرپایه‌ای** در نظر گرفته می‌شوند.

۱.۱.۱ دوگانی

هر مسأله‌ی LP به عنوان یک مسأله‌ی اولیه در نظر گرفته می‌شود که برای این نوع مسأله می‌توان یک مسأله‌ی متناظر تعریف کرد. مسأله‌ی متناظر با مسأله‌ی اولیه را مسأله‌ی دوگان می‌نامند. دو شکل (تعریف) مهم از دوگان وجود دارد: شکل استاندارد و شکل متعارفی. این اشکال کاملاً معادل هستند و به ترتیب از نمایش استاندارد و متعارفی مسائل برنامه‌ریزی خطی به دست می‌آیند.

[۱] Basic Feasible Solution

در هر مسأله‌ی دوگان، دقیقاً یک متغیر دوگان برای هر محدودیت اولیه و دقیقاً یک محدودیت دوگان برای هر متغیر اولیه وجود دارد. دوگان مسأله‌ی (۱.۱) را می‌توان به صورت

$$\begin{aligned} max \quad & w^T b \\ s.t. \quad & w^T A \leq c, \\ & w \;\; free. \end{aligned}$$

نمایش داد که در آن w متغیر دوگان متناظر با محدودیت خطی مسأله‌ی (۱.۱) است.

یک نتیجه‌ی مهم از دوگانی که به قضیه‌ی دوگانی معروف است، این است که هر جواب شدنی از دوگان یک مسأله‌ی LP، کرانی برای مقدار بهینه‌ی مسأله‌ی اولیه است. همچنین در برخی موارد حل مسأله‌ی دوگان ساده‌تر از حل مسأله‌ی اولیه است. در ادامه به برخی از قضایای موجود بین مسائل اولیه و دوگان اشاره می‌کنیم.

قضیه ۱.۱ (**قضیه‌ی دوگانی ضعیف** [۲]) اگر x جواب شدنی مسأله‌ی اولیه و w جواب شدنی مسأله‌ی دوگان باشد (در شکل متعارفی)، آنگاه

$$w^T b \leq c^T x.$$

به عبارت دیگر، مقدار تابع هدف به ازای هر جواب شدنی مسأله‌ی مینیمم سازی همواره بزرگتر یا مساوی مقدار تابع هدف به ازای هر جواب شدنی مسأله‌ی ماکزیمم سازی است.

قضیه ۲.۱ (**قضیه‌ی دوگانی قوی** [۲]) اگر مسأله‌ی برنامه‌ریزی خطی جواب بهینه داشته باشد، مسأله‌ی دوگان نیز جواب بهینه دارد و مقادیر توابع هدف بهینه‌ی آنها مساوی هستند.

قضیه ۳.۱ (**قضیه‌ی مکمل زائد** [۲]) فرض کنید x و w به ترتیب جواب‌های شدنی مسائل اولیه و دوگان باشند. بردارهای x و w جواب‌های بهینه‌ی مسائل متناظرشان هستند اگر و فقط اگر

$$\begin{aligned} w_i(a^i x - b_i) = \circ \qquad & \forall i, \\ (c_j - w^T a_j) x_j = \circ \qquad & \forall j. \end{aligned}$$

۲.۱.۱ روش‌های حل LP

معروف‌ترین روش حل برای مسائل LP روش **سیمپلکس**[۲] است. این روش توسط پروفسور جرج دانتزیگ[۳] در سال ۱۹۴۷ ابداع شد. سیمپلکس یک روش مرحله‌ای برای حل مسائل LP است. روش سیمپلکس متکی بر این مطلب است که اگر یک مسأله‌ی برنامه‌ریزی خطی جواب بهینه داشته باشد، آنگاه یک جواب شدنی اساسی وجود دارد که بهینه است [۲]. در روش سیمپلکس جستجو برای یافتن یک جواب بهینه با حرکت از یک جواب شدنی پایه‌ای به جواب شدنی پایه‌ای دیگر، در امتداد یال‌های مجموعه‌ی شدنی و همواره در جهت تقلیل دادن هزینه، انجام می‌پذیرد. در نهایت، یک جواب شدنی پایه‌ای به دست می‌آید که هیچ یک از یال‌های موجود در آن نمی‌توانند باعث کاهش هزینه شوند. چنین جواب شدنی پایه‌ای بهینه است و الگوریتم خاتمه می‌یابد. رفتن از یک جواب شدنی پایه‌ای به یک جواب شدنی پایه‌ای دیگر تا زمانی که الگوریتم به یک جواب بهینه برسد، یا با این نتیجه که مسأله جواب بهینه ندارد، ادامه می‌یابد. امروزه برای حل مسائل LP با استفاده از روش سیمپلکس، نرم افزارهای مختلفی طراحی شده است که از آن جمله می‌توان به نرم افزار $CPLEX$ اشاره کرد.

روش‌های حل دیگری که در اواسط دهه ۱۹۸۰ برای برنامه‌ریزی خطی ارائه شدند، روش‌های **نقطه درونی**[۴] هستند. این روش‌ها یک جواب بهینه را با حرکت در درون مجموعه‌ی شدنی پیدا می‌کنند، به این دلیل به آنها روش‌های نقطه درونی می‌گویند. در عمل، این روش‌ها قابل مقایسه با روش سیمپلکس هستند و اغلب، بویژه برای مسائل بزرگ و تنک، بهتر از روش سیمپلکس عمل می‌کنند. از نظر تئوری، روش‌های نقطه درونی منجر به الگوریتم‌های کارا (زمان چندجمله‌ای) می‌شوند و از نظر عملی، حل مسائل بزرگ را، که در اکثر کاربردها بروز می‌کنند، ممکن می‌سازند.

[۲] Simplex
[۳] George Dantzig
[۴] Interior-Point

۳.۱.۱ برنامه‌ریزی صحیح

برنامه‌ریزی خطی کاربردهای گوناگونی دارد. محدودیت عمده‌ای که از گستردگی چنین کاربردهایی می‌کاهد فرض بخش‌پذیری برنامه‌ریزی خطی است (فرض بخش‌پذیری به این معنی است که مقدار یک متغیر می‌تواند اعشاری باشد). لیکن، در بسیاری از مسائل واقعی، اگر مقادیری غیر از عدد صحیح به متغیرها داده شود، نتیجه بی‌معنا خواهد بود. به عنوان نمونه به متغیرهایی نظیر تعداد نفرات و تعداد دستگاه‌ها باید مقادیر صحیح نسبت داده شود. به این ترتیب، اگر تنها تفاوت مدل‌بندی مسأله با یک مسأله‌ی برنامه‌ریزی خطی، در نظر گرفتن محدودیت عدد صحیح باشد، به آن برنامه‌ریزی صحیح[۵] (IP) می‌گویند.

یک مسأله‌ی برنامه‌ریزی صحیح به فرم استاندارد به صورت

$$\begin{aligned} min \quad & c^T x \\ s.t. \quad & Ax = b, \\ & x \geq 0, \\ & x \ \ integer. \end{aligned} \tag{۱.۲}$$

نمایش داده می‌شود. گاهی ممکن است متغیرهای صحیح محدود به اعداد صفر و یک باشند، در این صورت متغیرها را **دودویی** می‌نامند. تفاوت یک مسأله‌ی برنامه‌ریزی صحیح با مسأله‌ی برنامه‌ریزی خطی در تعداد جواب‌های آن است. تعداد جواب‌های مسأله‌ی برنامه‌ریزی صحیح به مراتب کمتر است. حل مسائل برنامه‌ریزی خطی به مراتب آسانتر از حل مسائل برنامه‌ریزی صحیح است. موفق‌ترین الگوریتم‌های حل مسائل برنامه‌ریزی صحیح در صورت امکان از روش سیمپلکس استفاده می‌کنند. به این صورت که قسمت‌هایی از مسأله را با یک مسأله‌ی برنامه‌ریزی خطی متناظر با آن (یعنی مسأله‌ی برنامه‌ریزی صحیح که محدودیت‌های صحیح آن حذف شده باشد) مرتبط می‌سازند. چنین مسأله‌ای را اصطلاحاً **برنامه‌ریزی خطی آزاد شده**[۶] می‌نامند.

[۵] Integer Programming
[۶] Linear Programming-Relaxation

۲.۱ پیچیدگی محاسباتی

مسأله‌ی مورد بررسی در این پایان نامه یک مسأله‌ی بهینه‌سازی است و هر مسأله‌ی بهینه‌سازی Ψ به کمک چهار جزء زیر ترسیم می‌شود.

۱) یک مجموعه از ورودی I_Ψ. فرض می‌کنیم همه‌ی ورودی‌ها صحیح هستند.

۲) مجموعه‌ی همه‌ی جواب‌های شدنی $S_\Psi(x)$ برای یک نمونه‌ی $x \in I_\Psi$.

۳) یک تابع f_Ψ که با زوج (x,y) تعریف می‌شود، به طوری که $x \in I_\Psi$ و $y \in S_\Psi(x)$. برای هر زوج (x,y)، تابع f_Ψ یک عدد مثبت که مقدار تابع هدف به ازای زوج (x,y) است را مربوط می‌سازد. f_Ψ تابع هدف نامیده می‌شود.

۴) هدف مسأله، که ماکزیمم سازی یا مینیمم سازی است.

برای هر مسأله‌ی بهینه‌سازی می‌توان یک مسأله‌ی **تصمیم**[۷] اختصاص داد، یعنی یک مسأله که جواب آن بله یا خیر است. که تعریف آن به صورت زیر است.

تعریف ۲.۱ مسأله‌ی تصمیم Π مسأله‌ای است که همه‌ی نمونه‌های I_Π در آن به دو مجموعه‌ی Y_Π شامل نمونه‌های مثبت و N_Π شامل نمونه‌های منفی تقسیم می‌شود، هدف مسأله این است که برای هر نمونه‌ی $x \in I_\Pi$ نشان دهد که آیا $x \in Y_\Pi$ هست یا خیر.

به این ترتیب برای مسأله‌ی مینیمم سازی Ψ که در بالا آمد، می‌توان مسأله‌ی تصمیم زیر را در نظر گرفت: نمونه‌ی $x \in I_\Psi$ و یک f^*_Ψ را در نظر بگیرید. آیا یک جواب $s \in S_\Psi(x)$ به طوری که $f_\Psi(s) \leq f^*_\Psi$ وجود دارد؟

تعریف ۳.۱ زمان اجرا برای یک الگوریتم به عنوان ماکزیمم زمان انجام محاسبات توسط الگوریتم روی یک نمونه‌ی مسأله با اندازه‌ی مشخص تعریف می‌شود. زمان اجرا را اغلب با $O(g(\xi))$ نشان می‌دهند که در آن $g : \mathbb{N} \longmapsto \mathbb{R}^+$ یک تابع و ξ اندازه‌ی نمونه‌ی مسأله است. برای دو تابع $f : \mathbb{N} \longmapsto \mathbb{R}^+$ و $g : \mathbb{N} \longmapsto \mathbb{R}^+$ اگر اعداد

[۷] Decision Problem

مثبت $c, n_0 \in \mathbb{N}$ وجود داشته باشند به طوری که برای $n \geq n_0$ داشته باشیم $f(n) \leq cg(n)$، آنگاه می‌نویسیم $f(n) = O(g(n))$.

همان طور که گفته شد، مسائل تصمیم مسائلی هستند که خروجی آنها دارای دو حالت بله و خیر باشد. ورودی یک مسأله‌ی محاسباتی می‌تواند با یک رشته‌ی باینری متناهی s، کدگذاری شود. طول یک رشته‌ی s را با $|s|$ نشان می‌دهیم. یک مسأله‌ی تصمیم Π را با مجموعه‌ای از رشته‌ها مشخص می‌کنیم که به ازای این رشته‌ها، خروجی مسأله بله می‌شود. یک الگوریتم A برای یک مسأله‌ی تصمیم، یک رشته ورودی s را دریافت می‌کند و مقدار بله یا خیر را بر می‌گرداند که این مقدار را با $A(s)$ نشان می‌دهیم و می‌گوییم A مسأله‌ی Π را حل می‌کند اگر برای همه‌ی رشته‌های s، $A(s)$ دارای مقدار بله باشد اگر و تنها اگر $s \in \Pi$.

تعریف ۴.۱ ([۱۱]) الگوریتم A دارای زمان اجرای چندجمله‌ای است اگر یک تابع $p(.)$ وجود داشته باشد که برای هر رشته ورودی s، A در حداکثر $O(p(|s|))$ گام خاتمه یابد.

مسائلی که در زمان چندجمله‌ای قابل حل‌اند در کل مسائل ساده‌ای هستند که این مسائل در کلاس مسائل $\mathcal{P}$ قرار می‌گیرند. در این پایان نامه تمرکز ما روی مسائل سخت می‌باشد. یعنی مسائلی که الگوریتم زمان چندجمله‌ای برای حل آنها وجود ندارد.

تعریف ۵.۱ ([۱۱]) کلاس $\mathcal{P}$، مجموعه‌ی تمامی مسائل تصمیم است که برای آنها یک الگوریتم قطعی با زمان چندجمله‌ای وجود دارد.

اکنون ایده‌ی بررسی کارآمد یک جواب برای یک مسأله را مستقل از اینکه آیا مسأله به طور کارآمد قابل حل است، در نظر می‌گیریم. ساختار یک الگوریتم بررسی کننده برای یک مسأله‌ی Π، متفاوت از الگوریتم حل کننده‌ی مسأله است. به منظور بررسی یک جواب به یک رشته ورودی s، مجزا از یک رشته‌ی تصدیق t نیاز داریم که نشانگر بله بودن یک نمونه‌ی s از مسأله‌ی Π است. بنابراین تعریف زیر را داریم.

تعریف ۶.۱ ([۱۱]) الگوریتم C، یک تصدیق‌کننده‌ی کارآمد برای مسأله‌ی تصمیم Π است اگر:

۱. C یک الگوریتم زمان چندجمله‌ای با دو آرگومان ورودی s و t باشد.

۲. یک تابع چندجمله‌ای p وجود داشته باشد که برای هر رشته ورودی s، داشته باشیم: $s \in \Pi$ اگر و تنها اگر یک رشته ورودی t وجود داشته باشد که $|t| \leq p(|s|)$ و $C(s,t)$ دارای مقدار بله باشد.

تعریف ۷.۱ ([۱۱]) کلاس $\mathcal{NP}$، مجموعه‌ی تمامی مسائل تصمیم است که برای آنها یک تصدیق کننده‌ی کارآمد وجود دارد.

نکته ۴.۱ عمل جستجو برای یافتن رشته‌ی t که موجب یک تصدیق کننده‌ی کارآمد برای پذیرش s می‌شود، به عنوان یک جستجوی غیر قطعی روی فضای تصدیق‌های ممکن t در نظر گرفته می‌شود، بنابراین $\mathcal{NP}$، مجموعه‌ی تمامی مسائل تصمیم است که برای آنها یک الگوریتم غیرقطعی با زمان چندجمله‌ای وجود دارد.

نکته ۵.۱ $\mathcal{P} \subseteq \mathcal{NP}$. زیرا اگر برای مسأله‌ای الگوریتم قطعی با زمان چندجمله‌ای وجود داشته باشد، می‌توان همان الگوریتم را غیر قطعی هم در نظر گرفت.

تعریف ۸.۱ ([۲]) فرض کنید Π_1 و Π_2 دو مسأله‌ی تصمیم باشند. می‌گوییم مسأله‌ی Π_1 **تبدیل** مسأله‌ی Π_2 در زمان چند جمله‌ای است هر گاه یک الگوریتم زمان چندجمله‌ای وجود داشته باشد به طوری که با مفروض بودن یک نمونه‌ی x_1 از مسأله‌ی Π_1، خروجی آن یک نمونه‌ی x_2 از مسأله‌ی Π_2 باشد، با این خاصیت که $x_1 \in Y_{\Pi_1}$ اگر و تنها اگر $x_2 \in Y_{\Pi_2}$، یا به عبارت دیگر x_1 یک نمونه‌ی بله Π_1 است اگر و تنها اگر x_2 یک نمونه‌ی بله مسأله‌ی Π_2 باشد.

از تعریف بالا می‌توان نتیجه گرفت که یک الگوریتم زمان چندجمله‌ای برای Π_2 را می‌توان برای حل یک نمونه از Π_1 به کار برد. حال می‌توانیم یک مسأله‌ی $\mathcal{NP}$−کامل را به صورت زیر تعریف کنیم.

تعریف ۹.۱ ([۲]) یک مسأله‌ی تصمیم Π یک مسأله‌ی$\mathcal{NP}$−**کامل** است اگر Π در $\mathcal{NP}$ باشد و همه‌ی مسائل تصمیم در $\mathcal{NP}$ را بتوان در زمان چندجمله‌ای به Π تبدیل کرد.

به این ترتیب $\mathcal{P} = \mathcal{NP}$ زمانی برقرار است که بتوان ثابت کرد که هر مسأله‌ی $\mathcal{NP}$−کامل در $\mathcal{P}$ است. از طرفی، اگر بخواهیم نشان دهیم که یک مسأله Π در $\mathcal{NP}$−کامل است، باید نشان دهیم که Π در $\mathcal{NP}$ است و یک مسأله را