بسم الله الرحمن الرحیم

دانشکده ریاضی و علوم کامپیوتر

پایان‌نامه برای دریافت درجه کارشناسی ارشد در رشته ریاضی کاربردی
گرایش تحقیق در عملیات

حل مسئله مکانیابی هاب تک تخصیصی بدون ظرفیت با الگوریتم ممتیک

استاد راهنما

دکتر محمود امین طوسی

استاد مشاور

دکتر مهدی زعفرانیه

پژوهشگر:

فرشته حسنی پژمان

شهریور ۱۳۹۶

باسمه تعالی

فرم ارزشیابی و صورتجلسه دفاع از پایان‌نامه کارشناسی ارشد

فرم ۱۱۳-ت

جلسه دفاع از پایان‌نامه آقای /خانم **فرشته حسنی پژمان** دانشجوی رشته ریاضی کاربردی گرایش تحقیق در عملیات به شماره دانشجویی ۹۳۱۳۱۳۳۰۴۰ با عنوان:

حل مسئله مکانیابی هاب تک تخصیصی بدون ظرفیت با الگوریتم ممتیک

در مورخه ..... در دانشکده ریاضی و علوم کامپیوتر تشکیل و توسط هیات داوران مورد ارزشیابی قرار گرفت و نمره ..... برابر درجه ..... برای آن تعیین گردید .

به‌این ترتیب از این تاریخ آقای/ خانم فرشته حسنی پژمان به عنوان کارشناس ارشد در رشته مذکور شناخته می‌شود .

موارد ارزشیابی	موارد	حداکثر نمره	نمره کسب شده
۱- کیفیت نگارش	رعایت اصول نگارش انسجام در تنظیم بخشهای مختلف ،کیفیت تصاویر ، جداول و اشکال ،تنظیم فهرست ها ، منابع و ماخذ	٤	
۲- کیفیت علمی	بررسی تاریخچه و سابقه تجربی و نظری موضوع انسجام منطقی دربخش های مختلف پایان نامه ،ابتکار و نوآوری ،اهمیت و ارزش علمی پایان نامه ،استفاده از منابع معتبر و جدید ،کیفیت تجزیه و تحلیل یافته ها و نتیجه گیری ،روشن بودن روش کار ،هدف ها و فرضیه های تحقیق ،جدید بودن روش تحقیق	۱۰	
۳-کیفیت ارایه در جلسه دفاع	تسلط بر موضوع و بیان واضح وتفهیم آن ،توانایی در پاسخگویی به سوالات مطرح شده در جلسه ، رعایت زمان ارایه , روش ارائه	٤	
٤-ارزشیابی گزارشات	گزارش های دوره ای پیشرفت کار(حداقل ٤مورد)	۱	
٥- خروجی پایان نامه	مقاله مستخرج از پایان نامه: این نمره به صورت زیر اختصاص می یابد ۱) چکیده کنفرانسی هر مورد ۰/۲۵ نمره تا سقف۰/۵ نمره ۲) مقاله کامل در مجموع مقالات همایشهای معتبر یا مقاله در مجلات علمی-ترویجی معتبر پذیرفته شده یا چاپ شده هر مورد ۰/۵ نمره تا سقف ۱نمره ۳)مقاله پذیرفته شده یا چاپ شده در مجلات علمی پژوهشی معتبر ۱ نمره ٤)مقاله ارسال شدهبه مجلات علمی پژوهشی معتبرهر مورد ۲۵/. نمره تا سقف ۰/۵ نمره ٥) دستگاه ساخته شده دارای گواهی ثبت اختراع یا به سفارش سازمان ها تا سقف ۱ نمره ٦) دستگاه ساخته شده کاربردی که به تایید رئیس دانشکده رسیده باشد تا سقف ۰/۵ نمره	۱	
جمع			

درجه معادل کسب شده: (از ۱۹تا ۲۰ عالی☐ از۱۸تا ۱۸/۹۹بسیارخوب ☐ از ۱٦تا ۱۷/۹۹خوب ☐ از ۱٤ تا ۱۵/۹۹قابل قبول☐ کمتر از ۱٤ غیر قابل قبول☐)

مشخصات هیات دوران

ردیف	نام و نام خانوادگی	سمت	مرتبه علمی	محل کار	امضـــــــــــا
۱	دکتر محمود امین طوسی	استاد راهنما	استادیار	دانشگاه حکیم سبزواری	
۲	دکتر مهدی زعفرانیه	استاد مشاور	استادیار	دانشگاه حکیم سبزواری	
۳		استاد داور	استادیار	دانشگاه حکیم سبزواری	
۴		نماینده تحصیلات تکمیلی	استادیار	دانشگاه حکیم سبزواری	

امضا

مدیر گروه

امضا

رئیس دانشکده

سوگند نامه دانش آموختگان دانشگاه حکیم سبزواری

به نام خداوند جان و خرد کزین برتر اندیشه بر نگذرد

اینک که به خواست آفریدگار پاک، کوشش خویش و بهره گیری از دانش استادان و سرمایه‌های مادی و معنوی این مرز و بوم، توشه‌ای از دانش و خرد گردآورده‌ام، در پیشگاه خداوند بزرگ سوگند یاد می‌کنم که در به کارگیری دانش خویش، همواره بر راه راست ودرست گام بردارم. خداوند بزرگ، شما شاهدان، دانشجویان و دیگر حاضران را به عنوان داورانی امین گواه می گیرم که ازهمه دانش و توان خود برای گسترش مرزهای دانش بهره گیرم و از هیچ کوششی برای تبدیل جهان به جایی بهتر برای زیستن، دریغ نورزم. پیمان می‌بندم که همواره کرامت انسانی را در نظر داشته باشم و همنوعان خود را در هر زمان و مکان تا سر حد امکان یاری دهم. سوگند می خورم که در به کارگیری دانش خویش به کاری که با راه و رسم انسانی، آیین پرهیزگاری، شرافت و اصول اخلاقی برخاسته از ادیان بزرگ الهی، به ویژه دین مبین اسلام، مباینت دارد دست نیازم. همچنین در سایه اصول جهان شمول انسانی و اسلامی، پیمان می‌بندم از هیچ کوششی برای آبادانی و سرافرازی میهن و هم میهنانم فروگذاری نکنم و خداوند بزرگ را به یاری طلبم تا همواره در پیشگاه او و در برابر وجدان بیدار خویش و ملت سرافراز، بر این پیمان تا ابد استوار بمانم.

نام و نام خانوادگی: فرشته حسنی پژمان

تاریخ و امضا:

تأییدیه‌ی صحت و اصالت نتایج

باسمه تعالی

اینجانب فرشته حسنی پژمان به شماره دانشجویی ۹۳۱۳۱۳۳۰۴۰ دانشجوی رشته ریاضی کاربردی مقطع تحصیلی کارشناسی ارشد تأیید می‌نمایم که کلیه‌ی نتایج این پایان‌نامه حاصل کار اینجانب و بدون هرگونه دخل و تصرف است و موارد نسخه برداری شده از آثار دیگران را با ذکر کامل مشخصات منبع ذکر کرده‌ام. درصورت اثبات خلاف مندرجات فوق، به تشخیص دانشگاه مطابق با ضوابط و مقررات حاکم (قانون حمایت از حقوق مؤلفان و مصنفان و قانون ترجمه و تکثیر کتب و نشریات و آثار صوتی، ضوابط و مقررات آموزشی، پژوهشی و انضباطی ...) با اینجانب رفتار خواهد شد و حق هرگونه اعتراض درخصوص احقاق حقوق مکتسب و تشخیص و تعیین تخلف و مجازات را از خویش سلب می‌نمایم. در ضمن، مسؤولیت هرگونه پاسخگویی به اشخاص اعم از حقیقی و حقوقی و مراجع ذی صلاح (اعم از اداری و قضایی) به عهده ی اینجانب خواهد بود و دانشگاه هیچ گونه مسؤولیتی در این خصوص نخواهد داشت.

نام و نام خانوادگی: فرشته حسنی پژمان

تاریخ و امضا:

مجوز بهره برداری از پایان‌نامه

بهره برداری از این پایان‌نامه در چهارچوب مقررات کتابخانه و با توجه به محدودیتی که توسط استاد راهنما به شرح زیر تعیین می‌شود، بلامانع است:

☐ بهره برداری از این پایان‌نامه برای همگان بلامانع است.

☐ بهره برداری از این پایان‌نامه با اخذ مجوز از استاد راهنما، بلامانع است.

☐ بهره برداری از این پایان‌نامه تا تاریخ ..... ممنوع است.

استاد راهنما: دکتر محمود امین طوسی

تاریخ و امضا:

تقدیم به:

پدر و مادرم

و

همسرم

سپاس خداوندگار حکیم را که با لطف بی کران خود، آدمی را زیور عقل آراست.

در آغاز وظیفه خود می دانم از زحمات بی دریغ استاد راهنمای خود، جناب آقای دکتر ...، صمیمانه تشکر و قدردانی کنم که قطعاً بدون راهنمایی های ارزنده ایشان، این مجموعه به انجام نمی رسید.

از جناب آقای دکتر ... که زحمت مطالعه و مشاوره این رساله را تقبل فرمودند و در آماده سازی این رساله، به نحو احسن اینجانب را مورد راهنمایی قرار دادند، کمال امتنان را دارم.

همچنین لازم می دانم از گروه پارسی‌لاتک در پاسخگویی به مشکلات کاربران کمال قدردانی را داشته باشم.

در پایان، بوسه می زنم بر دستان خداوندگاران مهر و مهربانی، پدر و مادر عزیزم و بعد از خدا، ستایش می‌کنم وجود مقدس شان را و تشکر می‌کنم از خانواده عزیزم به پاس عاطفه سرشار و گرمای امیدبخش وجودشان، که بهترین پشتیبان من بودند.

فرشته حسنی پژمان

شهریور ۱۳۹۶

فهرست مطالب

فهرست جداول د

فهرست تصاویر ه

فهرست الگوریتم‌ها و

چکیده ۱

پیش‌گفتار ۲

فصل ۱: معرفی مسائل مکانیابی هاب ۴

۱-۱ مقدمه ۴

۱-۲ تاریخچه ۵

۱-۳ مفاهیم ۵

۱-۳-۱ شبکه تک هابی ۶

۱-۳-۲ شبکه چند هابی ۷

۱-۴ فرمولبندی مسائل مکانیابی هاب ۹

۱-۴-۱ مسئله مکانیابی تک- هاب ۱۰

۱-۴-۲ مسأله مکانیابی p - هاب ۱۱

۱-۴-۳ مسأله مکانیابی p - هاب میانه ۱۲

۱-۴-۴ مسأله مکانیابی p - هاب با هزینه ثابت پیوند ۱۳

۱-۴-۵ مسأله مکانیابی p - هاب با حداقل مقدار جریان روی پیوندها ۱۴

۱-۴-۶ مسأله مکانیابی p - هاب با ظرفیت محدود ۱۴

۱-۴-۷ مسأله مکانیابی p - هاب پیوسته ۱۵

۱-۴-۸ مسأله مکانیابی p - هاب چندهدفه ۱۶

۱-۴-۹ مسأله مکانیابی p - هاب مرکزی ۱۷

۱-۴-۱۰ مسأله مکانیابی p - هاب پوششی ۱۸
۱-۴-۱۰-۱ مسأله مکانیابی هاب مجموعه پوشش ۱۸
۱-۴-۱۰-۲ مسأله مکانیابی p - هاب حداکثر پوشش ۱۹
۱-۵ فرمولبندی مسأله مکانیابی هاب تک تخصیصی بدون ظرفیت ۲۰
۱-۶ رویکردهای حل مسأله ۲۱

فصل ۲: کاربرد الگوریتم جستجوی ممنوعه و الگوریتم ژنتیک در حل مسائل مکانیابی هاب ۲۳

۲-۱ مقدمه‌ای بر الگوریتم‌های بهینه‌سازی فراابتکاری ۲۳
۲-۱-۱ مسائل بهینه‌سازی ۲۳
۲-۱-۲ پیچیدگی محاسباتی ۲۴
۲-۱-۳ روش‌های حل مسائل بهینه‌سازی ۲۶
۲-۱-۴ الگوریتم‌های فراابتکاری ۲۷
۲-۱-۴-۱ الگوریتم‌های فراابتکاری مبتنی بر یک جواب ۲۹
۲-۱-۴-۲ الگوریتم‌های فراابتکاری مبتنی بر جمعیت ۳۱
۲-۲ الگوریتم جستجوی ممنوعه (TS) ۳۳
۲-۲-۱ مقدمه و تاریخچه ۳۳
۲-۲-۲ ساختارهای حافظه و تابو ۳۳
۲-۲-۳ معیارهای توقف ۳۸
۲-۲-۴ ساختار کلی جستجوی ممنوعه ۳۸
۲-۲-۵ استراتژی‌های پیشرفته‌ی الگوریتم جستجوی ممنوعه ۴۲
۲-۲-۶ جستجوی ممنوعه برای حل مسأله USAHLP ۴۳
۲-۳ الگوریتم ژنتیک (GA) ۵۲
۲-۳-۱ مفاهیم پایه‌ای ۵۲
۲-۳-۲ استراتژی‌های انتخاب ۵۳
۲-۳-۲-۱ انتخاب مبتنی بر تناسب برازندگی ۵۴
۲-۳-۲-۲ انتخاب وصفی ۵۵
۲-۳-۳ عملگرهای تغییر ۵۶
۲-۳-۳-۱ عملگر تقاطع ۵۷
۲-۳-۳-۲ عملگر جهش ۵۸
۲-۳-۴ جایگزینی ۶۰

۲-۳-۵ ساختار کلی GA ۶۰
۲-۳-۶ الگوریتم ژنتیک برای حل مسأله USAHLP ۶۴

فصل ۳: طرح یک الگوریتم ممتیک کارامد **۶۹**
۳-۱ حل مسئله مکانیابی با الگوریتم ممتیک ۷۰
۳-۱-۱ مقداردهی اولیه الگوریتم ممتیک ۷۲
۳-۱-۲ جستجوهای محلی ابتکاری HeurLoc و HeurAlloc ۷۴

فهرست منابع **۷۹**

فهرست جداول

۱-۱ طبقه‌بندی مسائل مکانیابی هاب [۱۶] ۹

۲-۱ کاربردهای جستجوی ممنوعه [۱] ۳۴

۲-۲ داده‌های مربوط به مسأله نمونه زمان‌بندی ۳۹

۲-۳ مقایسه‌ی مفاهیم بهینه‌سازی و فرایند الگوریتم ژنتیک ۵۳

۲-۴ مقادیر برازندگی جمعیتی با اندازه $n = ۵$ ۵۶

۲-۵ مسأله کوله‌پشتی با ۱۰ شی ۶۲

۲-۶ جمعیت اولیه تولید شده ۶۲

۲-۷ فرزندان تولید شده در تکرار ۱ . . . . . ۶۳

۲-۸ فرزندان تولید شده در تکرار ۱ بعد از عمل جهش ۶۴

۲-۹ جمعیت تولید شده بعد از تکرار ۱ . . . . . ۶۴

۳-۱ معرفی پارامترها ۷۵

فهرست تصاویر

۱-۱ شبکه کامل با ۶ گره ۶
۱-۲ شبکه تک هابی با ۶ گره ۶
۱-۳ شبکه چند هابی با ۱۵ گره ۷
۱-۴ مسیرهای جایگزین بین مبدأ i و مقصد j ۸
۱-۵ نمایی کامل از شبکه هاب تک تخصیصی ۲۲

۲-۱ رابطه بین مسائل کلاس P ، NP ، NP-Complete و NP-hard ۲۶
۲-۲ الگوریتم‌های مبتنی بر یک جواب ۲۸
۲-۳ الگوریتم‌های مبتنی بر جمعیت ۲۸
۲-۴ به دام افتادن الگوریتم در جواب بهینه محلی (x_n) ۳۱
۲-۵ ساختار حافظه در الگوریتم TS ۳۶
۲-۶ نحوه‌ی شکل‌گیری لیست ممنوع ۳۷
۲-۷ انتخاب چرخ رولت ۵۵
۲-۸ انتخاب جامع احتمالی ۵۶
۲-۹ تقاطع تک نقطه‌ای ۵۸
۲-۱۰ تقاطع دو نقطه‌ای ۵۸
۲-۱۱ تقاطع یکنواخت ۵۹
۲-۱۲ جهش ژنی ۵۹
۲-۱۳ جهش تعویض ۶۰
۲-۱۴ احتمال انتخاب کروموزم‌ها با استفاده از چرخ رولت ۶۳
۲-۱۵ بکارگیری عملگر تقاطع ۶۶
۲-۱۶ نمودار نمایش مقدار تنوع جمعیت با تغییر شماره نسل‌ها ۶۷

۳-۱ شبکه ساده با ۴ گره (رنگ سیاه معرف گره هاب است.) ۷۳

فهرست الگوریتم‌ها

۲-۱ شبه کد الگوریتم‌های فراابتکاری مبتنی بر یک جواب ۲۹
۲-۲ شبه کد جستجوی محلی ۳۰
۲-۳ شبه کد الگوریتم‌های فراابتکاری مبتنی بر جمعیت ۳۲
۲-۴ شبه کد الگوریتم جستجوی ممنوع ۳۹
۲-۵ روش TabuAlloc [۲] ۴۶
۲-۶ ایجاد جواب اولیه با روش DROP ۴۷
۲-۷ الگوریتم TSUSAHLP ۵۱
۲-۸ شبه کد الگوریتم چرخ رولت ۵۵
۲-۹ شبه کد الگوریتم ژنتیک ۶۱
۳-۱ ساختار جستجوی محلی HeurLoc ۷۷
۳-۲ ساختار جستجوی محلی HeurAlloc ۷۸

فرم چکیده ی پایان‌نامه ی دوره ی تحصیلات تکمیلی

مدیریت تحصیلات تکمیلی

دانشگاه حکیم سبزواری

نام خانوادگی دانشجو:حسنی پژمان	نام:فرشته	ش. دانشجویی:۹۳۱۳۱۳۳۰۴۰
استاد راهنما: دکتر محمود امین طوسی		
استاد مشاور: دکتر مهدی زعفرانیه		
دانشکده ریاضی و علوم کامپیوتر	رشته: ریاضی کاربردی	گرایش: تحقیق در عملیات
مقطع: کارشناسی ارشد	تاریخ دفاع:شهریور ۱۳۹۶	تعداد صفحات: ۸۲
عنوان پایان‌نامه: حل مسئله مکانیابی هاب تک تخصیصی بدون ظرفیت با الگوریتم ممتیک		
کلید واژه‌ها:الگوریتم‌های ممتیک، طراحی شبکه، مسائل مکانیابی هاب، جستجوی محلی		

چکیده: در دهه‌های گذشته، علم مکانیابی به حوزه پژوهشی بسیار فعالی تبدیل شده که توجه بسیاری از محققان و پژوهشگران را به خود جلب کرده‌است. مسائل مکانیابی تسهیلات که در هسته‌ی این رشته علمی قرار گرفته‌اند، دارای زیرمجموعه‌ای بسیار کاربردی و مهم با عنوان مسائل مکانیابی هاب هستند که در دسته‌ی مسائل سخت طبقه‌بندی می‌شوند. در اکثر زمینه‌های تحقیقاتی، می‌توان مسئله را به‌صورت یک مسأله بهینه‌سازی بیان کرد. با افزایش اندازه و پیچیدگی مسائل بهینه‌سازی، الگوریتم‌های فراابتکاری روز به روز مهمتر می‌شوند. یکی از نشانه‌های این اهمیت، تعداد زیاد کنفرانس‌ها، مجلات و کتاب‌هایی است که در مورد طراحی و کاربرد الگوریتم‌های فراابتکاری منتشر شده‌اند. در سال‌های اخیر، ترکیب الگوریتم‌های فراابتکاری با یکدیگر، بسیار رایج شده‌است به گونه‌ای که بهترین نتایج برای مسائل بهینه‌سازی دنیای واقعی و یا مسائل کلاسیک، به‌وسیله‌ی الگوریتم‌های ترکیبی بدست آمده‌اند. الگوریتم‌های ممتیک محصول ترکیب دو الگوریتم فراابتکاری یعنی الگوریتم‌های تکاملی و الگوریتم‌های جستجوی محلی هستند. در این پایان‌نامه یک الگوریتم ممتیک کارا و مؤثر برای حل یکی از انواع مسائل مکانیابی هاب به نام مسئله مکانیابی هاب تک تخصیصی بدون ظرفیت مورد بررسی قرار گرفته است.

پیش‌گفتار

مهندسان و تصمیم‌گیرندگان، با مسائلی روبرو هستند که پیچیدگی آن‌ها روز به روز افزایش پیدا می‌کند. این مسائل در زمینه‌های مختلفی نظیر تحقیق در عملیات، طراحی سیستم‌های مکانیکی، پردازش تصویر و الکترونیک ایجاد می‌شوند[۱]. علم مکانیابی، رشته‌ای پربار و غنی است که انواع زیادی از مسائل را در برمی‌گیرد. تحقیقات انجام شده در این زمینه منجر به تولید حجم قابل توجهی دانش هم از نظر خصوصیات تئوریکی و هم چارچوب مدلسازی همراه با تکنیک‌های حل شده‌است. در سال‌های اخیر مطالعات مکانیابی به عنوان یکی از عناصر کلیدی در موفقیت و بقای مراکز صنعتی مطرح است. این دانش در طول زمان تکامل یافته، با احساس نیاز به حل مسائل مکانیابی کاربردی، به خاطر چالش‌های فنی و نظری و اغلب بواسطه مشکلات برخواسته در رشته‌های مختلف، تحقیقات زیادی را به خود معطوف داشته است. در واقع، تعامل با سایر رشته‌ها مانند اقتصاد، جغرافیا، علم منطقه‌ای و لاجستیک [۱] و... همواره یک نیروی محرکه توسعه علم مکانیابی بوده‌است که امروزه پتانسیل این زمینه مطالعاتی در بسیاری از سیستم‌های دنیای واقعی به‌طور گسترده‌ای به رسمیت شناخته می‌شود. مسائل مکانیابی تسهیلات (FLPs)[۲] در هسته این رشته علمی قرار دارد که به دنبال تعیین بهترین مکان برای یک یا چند تسهیل در جهت خدمت‌دهی به مجموعه‌ای از نقاط تقاضا است [۳].

اساسی‌ترین موضوع در سیستم‌های حمل و نقل و شبکه‌های ارتباطی، ارسال جریان از مبدأ به مقصد است و از این‌رو تحقیقات انجام گرفته همواره در جهت سهولت و کاهش هزینه‌ی ارتباطی بین نقاط مبدأ و مقصد پیش رفته است. در این راستا مسئله طراحی شبکه هاب که زیرمجموعه‌ای از مسائل مکانیابی تسهیلات است، با اینکه زمان کمی از فعالیتش می‌گذرد جایگاه ویژه‌ای در میان مسائل مکانیابی یافته است. در این مسئله، جریان از یک سری منبع به یک سری مقصد ارسال می‌شود، اما از آنجایی که ارتباط مستقیم هر کدام با دیگری هزینه بسیار زیادی دارد یا امکان‌پذیر نیست، تعدادی از مراکز به عنوان نقاط هاب در نظر گرفته می‌شوند. انتخاب مکان بهینه‌ی هاب‌ها و همچنین تخصیص مناسب نقاط مبدأ و مقصد به هاب‌ها، طوریکه هزینه‌ی حمل و نقل و تأسیس هاب‌ها به کمترین مقدار خود برسد، هدف نهایی مسئله است.

این پایان‌نامه شامل ۴ فصل است:

در فصل ۱ تاریخچه، مفاهیم مورد نیاز، انواع مسائل مکانیابی هاب و روش‌های حل این دسته از مسائل مطرح خواهند شد.

در فصل ۲ بعد از تشریح الگوریتم‌های فراابتکاری جستجوی ممنوعه و ژنتیک، کاربرد این الگوریتم‌ها در حل مسائل

---

[۱] logistic

[۲] Facility Location Problems

مکانیابی هاب بیان می‌شود.

در فصل ۳ الگوریتم‌های ممتیک شرح داده می‌شوند و برای نوع خاصی از مسائل موردبحث در این پایان‌نامه رویکرد حلی با استفاده از این الگوریتم‌ها ارائه خواهد شد.

مطالب این پایان‌نامه برگرفته از مقالات زیر می‌باشد:

1. Abyazi Sani, Roya and Ghanbari, Reza. An efficient tabu search for solving the uncapacitated single allocation hub location problem. *Comput. Ind. Eng.*, 93(c):99–109, 2016.

2. Marić, M., Stanimirović, Z., and Stanojević, P. An efficient memetic algorithm for the uncapacitated single allocation hub location problem. *Soft Comput.*, 17(3):445–466, 2013.

3. Farahani, R., Hekmatfar, M., Boloori Arabani, A., and Nikbakhsh, E. Hub location problems: A review of models, classification, solution techniques, and applications. *Comput. Ind. Eng.*, 64(4):1096–1109, 2013.

4. Topcuoglu, H., Corut, F., Ermis, M., and Yilmaz, G. Solving the uncapacitated hub location problem using genetic algorithms. *Comput. Oper. Res.*, 32(4):967–984, 2005.

فصل ۱

معرفی مسائل مکانیابی هاب

۱-۱ مقدمه

مسئله مکانیابی هاب (HLP)[۱] از موضوعات جدید در زمینه مسائل مکانیابی است. شبکه‌های کامپیوتری، مخابراتی و حمل‌ونقل که دارای نقاط مبدأ / مقصد (O/D)[۲] بسیاری برای انتقال جریان هستند، معمولاً از ساختارهای هاب-غیرهاب[۳] برای تعیین کارآمد مسیر جریان بین این نقاط بهره می‌گیرند. در حقیقت، این ساختار در سیستم‌هایی که ارتباط مستقیم بین تمام نقاط امکان‌پذیر نیست و یا پرهزینه است، استفاده می‌شود. ویژگی کلیدی این ساختارها، در مقایسه با شبکه‌های کلاسیک، امکان اتصال تعداد زیادی جفت‌های مبدأ و مقصد با تعداد لینک‌های کمتر با استفاده از گره‌هایی است که تسهیلات هاب[۴] نامیده می‌شوند. در این پایان‌نامه، هرچیزی که بین دو نقطه در شبکه جابجا می‌شود اعم از داده‌های مخابراتی، کالا، مسافر و ... جریان نامیده می‌شود. به‌منظور کاهش هزینه تأسیس هاب‌ها، تمرکزدهی به عملیات بررسی و جداسازی کالاها و دستیابی به صرفه‌ی اقتصادی هزینه‌های مسیریابی، جریان‌های دارای مبدأ یکسان اما مقصدهای متفاوت که به سوی هاب‌ها تعیین مسیر شده‌اند، گردآوری می‌شوند و سپس با جریان‌های دارای مبدأهای متفاوت و مقصد یکسان ترکیب می‌گردند. مسائل مکانیابی هاب، شامل تعیین مکان تسهیلات هاب و طراحی شبکه‌های هاب به‌منظور بهینه‌سازی تابع هدف مبتنی بر هزینه (یا مبتنی بر خدمت) می‌باشند. دشواری اصلی این دسته از مسائل که مربوط به ایجاد پیوند بین تصمیمات مربوط به تعیین محل هاب‌ها و طراحی شبکه می‌شود، ریشه در وابستگی‌های ذاتی میان دو سطح از فرآیند تصمیم‌گیری دارد. سطح اول به گزینش مجموعه‌ای از گره‌ها برای تعیین محل تسهیلات هاب می‌پردازد درحالی‌که سطح دوم با انتخاب لینک‌هایی برای اتصال مبدأها، مقصدها و هاب‌ها و همچنین تعیین مسیر جریان‌ها از طریق شبکه، با طراحی شبکه هاب سروکار دارد. در ادامه به بیان تاریخچه، مفاهیم، طبقه‌بندی و روش‌های حل این نوع از مسائل می‌پردازیم.

---

[۱] Hub Location Problem [۲] Origin/Destination [۳] hub-spoke [۴] hub facilities

۱-۲ تاریخچه

مسئله مکانیابی هاب، تاریخچه کوتاهی دارد. اولین مقاله در این زمینه در سال ۱۹۸۵ [۴] منتشر شد که درباره‌ی کاربرد این مسئله در خطوط هوایی و فرودگاه‌ها بوده‌است. اگرچه حکیمی [۵] اولین کسی بود که مقاله‌ای راجع به مکانیابی هاب منتشر کرد، اما با توجه به اینکه مقاله بعدی در باب این موضوع تا دو دهه‌ی بعد منتشر نشد، این‌طور تصور می‌شود که مسئله مکانیابی هاب برای اولین بار در دهه‌ی ۸۰ مطرح شده‌است. اُکلی [۱] مدل‌های مکانیابی هاب را توسعه داد. وی به بررسی مدل‌های پیوسته [۶] و مدل‌های گسسته [۷، ۸] مسائل مکانیابی هاب پرداخت و بعد از آن ناحیه تحقیقاتی وسیعی به این دسته از مسائل اختصاص یافت. در طی سه دهه اخیر پژوهشگرانی از سراسر جهان با زمینه‌های تحقیقاتی مختلف همچون علوم مکانیابی، جغرافیایی، منطقه‌ای، علوم کامپیوتری، بهینه‌سازی شبکه، حمل‌ونقل و مخابرات به مطالعاتی در این حوزه پرداخته‌اند.

مقالات مروری و بررسی‌های متعددی در حوزه مسائل مکانیابی هاب وجود دارند که هر کدام روی جنبه‌های مختلفی از این مسائل متمرکز شده‌اند. اولین مقالات مروری در این زمینه توسط اُکلی و میلر [۲] [۹] و کمپبل [۳] [۱۰] ارائه شد. این مقالات برای مدل‌های بنیادی و همچنین ساختارهای توپولوژیکی قابل استفاده شبکه‌های هاب، طبقه‌بندی مناسبی فراهم کردند. کلینسویکز [۴] [۱۱] روی طراحی شبکه‌های هاب با محوریت شبکه‌های مخابراتی متمرکز شد و برایان [۵] و اُکلی [۱۲] یک بررسی روی شبکه‌های حمل‌ونقل هوایی انجام دادند. کمپبل و همکاران [۱۳] بررسی جامعی روی مسائل مکانیابی هاب با تصمیم شاخص تعیین محل هاب‌ها ارائه کردند و آلومور [۶] و کارا [۷] [۱۴] ادبیات در حال رشد مدل‌های مکانیابی هاب شبکه‌ای تا قبل از سال ۲۰۰۸ را مورد بررسی و طبقه‌بندی قرار دادند. ارائه نگرشی در قالب انگیزه‌های اولیه‌ی تجزیه و تحلیل مسائل مکانیابی هاب توسط کمپبل و اُکلی [۱۵] فراهم گردید که جهت تحقیقات اخیر نیز تبیین شد و در نهایت زنجیرانی فراهانی و همکاران [۱۶] روش‌های حل و کاربردهای چندین نوع از مسائل مکانیابی هاب را مرور کردند.

۱-۳ مفاهیم

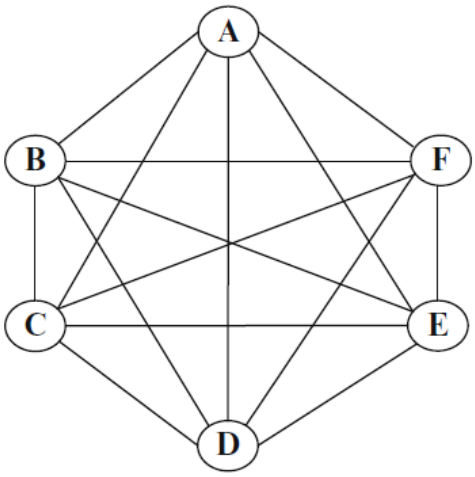
در این بخش، درباره‌ی خدماتی همچون جابجایی اطلاعات، کالاها و افرادی که بین یک جفت مبدأ و مقصد واقع شده‌اند بحث خواهیم کرد[۱۷]. هر جفت مبدأ و مقصد به خدمتی متفاوت از جفت گره‌های دیگر نیازمند است. بنابراین، کالاهای حمل شده از مبدأ i به مقصد j قابل معاوضه با کالاهای حمل شده از مبدأ j به مقصد i نخواهند بود.

تعریف ۱-۳-۱. شبکه‌ای که تمام گره‌های آن به یکدیگر متصل باشند، شبکه کامل [۸] نامیده می‌شود.

فرض کنید n گره داریم به‌طوری که هر گره دارای این قابلیت است که بتواند به‌عنوان یک گره مبدأ یا مقصد در نظر گرفته شود. در این‌صورت، شبکه‌ای کامل با $n(n-1)$ جفت گره خواهیم داشت که جفت گره (i,j) متفاوت از جفت گره (j,i)

[۱] O'Kelly [۲] Miller [۳] Campbell [۴] Klincewicz [۵] Bryan [۶] Alumur [۷] Kara [۸] fully connected network

است. یک شبکه کامل با ۶ گره و۳۰ جفت مبدأ و مقصد در شکل ۱-۱ نشان داده شده‌است.

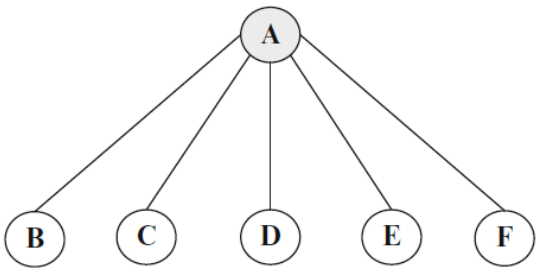


شکل ۱-۱: شبکه کامل با ۶ گره

تعریف ۱-۳-۲. هاب در لغت به گلوله‌ای اطلاق می‌شود که در وسط یک چرخ قرار گرفته است و کمپبل [۱۸] هاب‌ها را تسهیلاتی معرفی کرد که به‌عنوان گره‌های انتقال و توزیع، به مبدأ و مقصدهای بسیاری سرویس‌دهی می‌کنند و در شبکه‌های ترافیکی و مخابراتی به‌کار می‌روند.

۱-۳-۱ شبکه تک هابی

اگر یکی از گره‌های شبکه را هاب در نظر بگیریم و آن را به گره‌های دیگر که غیرهاب [۱] یا اسپوک نامیده می‌شوند وصل کنیم، در این‌صورت برای سرویس همه‌ی مبدأها و مقصدها تعداد $۲(n-۱)$ لینک وجود خواهد داشت که این شبکه در شکل ۱-۲ نشان داده شده‌است. اشکال اصلی چنین سیستمی که به همه‌ی گره‌ها جز گره هاب مربوط می‌شود، الزام عبور از گره



شکل ۱-۲: شبکه تک هابی با ۶ گره

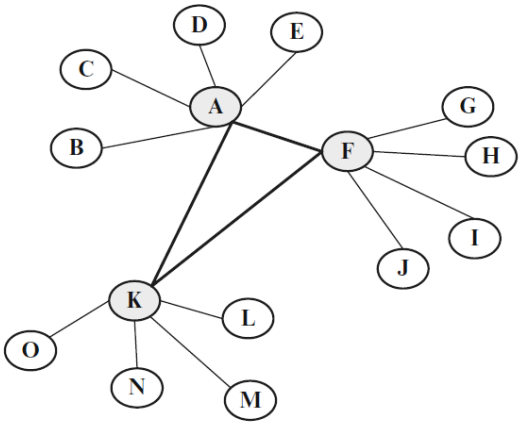
هاب برای رفتن از هر گره غیرهاب به گره غیرهاب دیگر است که منجر به ایجاد بیش از یک گردش برای طی هر جفت مبدأ

[۱] spoke

و مقصد می‌شود. با فرض اینکه ترافیک سرویس متفاوتی در شبکه وجود داشته باشد و هر وسیله نقلیه بتواند روزانه ۵ جفت مبدأ و مقصد را سرویس‌دهی کند، در این‌صورت با ۱۸ وسیله نقلیه می‌توان روزانه ۹۰ جفت را سرویس‌دهی کرد. بنابراین، با توجه به معادله $n(n-1)$ قادر به سرویس ۱۰ گره در هر روز خواهیم بود. این در حالی است که با همین تعداد وسیله در شبکه تک هابی با توجه به معادله $2(n-1)$ می‌توان روزانه ۴۶ گره را سرویس‌دهی کرد. در نتیجه با وسایل نقلیه ثابت، تعداد شهرهای بیشتری در شبکه هاب نسبت به شبکه کامل سرویس‌دهی می‌شوند که در استدلال فوق ظرفیت وسایل نقلیه نادیده گرفته شده‌است.

۱-۳-۲ شبکه چند هابی

در شبکه‌های چند هابی، فرض بر این است که گره‌های هاب کاملاً به یکدیگر متصل شده‌اند و هر گره غیرهاب یا دقیقاً با یک گره هاب در ارتباط است که معرف شبکه تک تخصیصی است و یا حداقل با یک گره هاب در ارتباط است که در این‌صورت معرف شبکه چندتخصیصی خواهد بود. همان‌طور که در شکل ۱-۳ می‌بینید، شبکه‌ای با ۱۵ شهر و ۳ هاب به تصویر کشیده شده‌است.



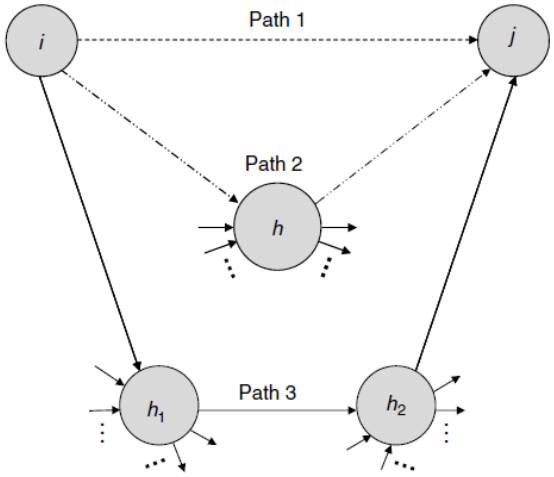
شکل ۱-۳: شبکه چند هابی با ۱۵ گره

معمولاً تعداد مسافران یا کالاهایی که از یک گره هاب به گره هاب دیگر حمل می‌شوند بیشتر از تعداد مسافران یا کالاهایی است که از هر گره غیرهاب به گره هاب موردنظر نقل مکان می‌کنند. برای مثال، اگر ترافیک بین هر جفت مبدأ و مقصد ۱۰ واحد باشد آنگاه ۱۴۰ واحد ترافیک بین هر گره غیرهاب و گره هاب اختصاص یافته به آن وجود خواهد داشت اما ترافیک بین دو گره هاب ۲۵۰ واحد خواهد بود.

در شبکه‌های رابط، مانند شبکه راه یا شبکه کامپیوتری، ایجاد مسیرهای مستقیم بین هر جفت گره کار معقولی نیست. گره‌های هاب برای حل این مشکل استفاده می‌شوند. تأسیس مسیرهایی با کیفیت بالا بین هاب‌ها باعث کسب منافع اقتصادی می‌شود که یکی از مزیت‌های استفاده از هاب‌هاست. در شبکه‌های راه‌آهن زمانی‌که بین دو گره تعداد زیادی راه ارتباطی وجود داشته باشد، ایجاد یک بزرگراه با چندین جاده بین این دو گره کاری اقتصادی است و منجر به حرکت سریع‌تر وسایل

نقلیه و اتلاف کمتر سوخت می‌شود و بنابراین در زمان و هزینه صرفه‌جویی می‌شود. در شبکه‌های کامپیوتری، کابل‌های فیبرنوری فقط برای اتصال هاب‌ها استفاده می‌شوند و به‌کاربردن آن‌ها برای اتصال هر دو گره کاری اقتصادی نیست.

شکل ۱-۴ سه مسیر ممکن از مبدأ i به مقصد j را نشان می‌دهد. در این تصویر، مسیر (۱) یک سیر مستقیم و یا



شکل ۱-۴: مسیرهای جایگزین بین مبدأ i و مقصد j

به‌عبارتی، پیکان مستقیم نقطه به نقطه از i به j ، مسیر (۲) سیری یک توقفه توسط هاب h و مسیر (۳) سیری دو توقفه توسط هاب‌های h_1 و h_2 است. در حقیقت، سه نقش عمده تسهیلات هاب عبارتند از:

۱. فاز جمع آوری [۱] : جریان از نقاط غیرهاب به نقاط هاب مربوطه جمع‌آوری می‌شود.

۲. فاز انتقال [۲] : جریان در صورت نیاز به دیگر هاب‌های شبکه، انتقال می‌یابد.

۳. فاز توزیع [۳] : تقاضاهای دریافت شده از دیگر هاب‌ها به نقاط غیرهاب مقصد ارسال می‌شود.

ویژگی‌های کلیدی مسائل مکانیابی هاب را می‌توان به‌صورت زیر خلاصه نمود:

- تقاضای سرویس با جریان بین جفت‌های مبدأ و مقصد در ارتباط است؛
- تسهیلات هاب، گره‌های میانی در مسیرهای مبدأ و مقصداند و به‌عنوان نقاط انتقال یا ترکیب عمل می‌کنند؛
- مسیریابی جریان ازطریق هاب‌ها دارای مزیت (یا الزام) است؛
- تابع هدف، مبتنی بر هزینه (یا مبتنی بر خدمت) است که به طراحی شبکه هاب (محل هاب‌ها و انتخاب پیوندها) و مسیریابی جریان بستگی دارد.

---

[۱] collection [۲] transfer [۳] distribution

۱-۴ فرمولبندی مسائل مکانیابی هاب

انواع مختلفی از مسائل مدل‌سازی هاب توسط محققان مورد بررسی قرار گرفته‌اند. با این حال، با توجه به جدول ۱-۱ و ترکیب موارد ذکر شده در آن، مدل‌های متنوعی از این مسائل مدل‌سازی ایجاد می‌شوند که هنوز تحقیقی بر روی آن‌ها انجام نگرفته‌است. از طرفی برخی از مدل‌های جدید دارای هیچ کاربردی در دنیای واقعی نیستند و ارزش مدل‌سازی ندارند. قابل ذکر است که برنامه‌های کاربردی مسائل هاب در جهان واقعی، اغلب گسسته هستند. لذا، مدل‌هایی که تاکنون توسعه یافته‌اند غالباً مدل‌های گسسته‌اند.

جدول ۱-۱: طبقه‌بندی مسائل مکانیابی هاب [۱۶]

حوزه جواب	شبکه (دامنه گره‌های هاب کاندید، همه‌ی گره‌های شبکه است.)
	گسسته (بخشی از گره‌های خاص به‌عنوان دامنه گره‌های هاب کاندید در نظر گرفته می‌شوند.)
	پیوسته (دامنه گره‌های هاب یک صفحه یا کره است.)
ضابطه	mini-max (حداکثر هزینه حمل‌ونقل از گره‌های مبدأ به گره‌های مقصد مینیمم می‌شود.)
	mini-sum (هزینه کل تعیین مکان گره‌های هاب و تخصیص گره‌های غیرهاب به گره‌های هاب مینیمم می‌شود.)
منبع تعیین تعداد هاب‌ها	برونی (تعداد هاب‌ها در ابتدای مسأله مشخص می‌شود.)
	درونی (تعداد هاب‌ها به‌عنوان قسمتی از جواب تعیین می‌شود.)
تعداد نقاط هاب	تک هابی
	چند هابی
ظرفیت هاب	محدود
	نامحدود
هزینه مکانیابی نقاط هاب	بدون هزینه
	هزینه ثابت
	هزینه متغیر
تخصیص گره غیرهاب به هاب	یک هاب (تک تخصیصی)
	بیش از یک هاب (چند تخصیصی)
هزینه اتصال گره غیرهاب به هاب	بدون هزینه
	هزینه ثابت
	هزینه متغیر

در ادامه به رایج‌ترین فرمول‌بندی‌های مسائل مکانیابی هاب که به‌طور گسترده در ادبیات این‌گونه مسائل اعمال می‌شوند، به‌طور خلاصه اشاره می‌کنیم [۱۶].

۱-۴-۱ مسئله مکانیابی تک- هاب [۱]

این مسئله اولین بار توسط اُکلی [۸] بیان شد که در آن ضابطه mini-sum است، حوزه جواب شبکه است، تعداد گره‌های هاب که قرار است تعیین مکان شوند به‌طور برونی تعریف شده و برابر با ۱ است، گره‌های غیرهاب به گره هاب منفرد متصل شده‌اند، تأسیس گره هاب هزینه‌ای ندارد، ظرفیت هاب نامحدود است و از آنجایی که تنها یک هاب تعیین مکان می‌شود مسئله تک تخصیصی خواهد بود. ورودی‌های مسأله به شرح زیر است:

W_{ij}: مقدار جریان بین گره‌های i و j

C_{ij}: هزینه واحد انتقال جریان از گره غیرهاب i به گره هاب j

خروجی‌های مدل باینری‌اند. Y_{ij} برابر با ۱ است اگر گره i به گره هاب j تخصیص داده شده باشد و در غیر این‌صورت برابر ۰ است. همچنین، Y_{jj} برابر با ۱ به این معنی است که گره j به خودش تخصیص یافته و در حقیقیت یک گره هاب است. با توجه به مطالب گفته شده، فرمول ریاضی این مسئله به‌صورت زیر بیان می‌شود:

$$\min \quad \sum_i \sum_j \sum_k W_{ik}(C_{ij} + C_{jk}) Y_{ij} Y_{kj} \tag{۱-۱}$$

$$s.t.$$

$$\sum_j Y_{jj} = ۱ \tag{۱-۲}$$

$$Y_{ij} - Y_{jj} \leqslant ۰ \qquad \forall i, j \tag{۱-۳}$$

$$Y_{ij} \in \{۰, ۱\} \qquad \forall i, j \tag{۱-۴}$$

معادله ۱-۱، هزینه‌ی کل انتقال جریان توسط هاب را به حداقل می‌رساند. معادله ۱-۲، تصریح می‌کند که فقط یک هاب وجود دارد. معادله ۱-۳، مشخص می‌کند که گره i تنها می‌تواند با یک هاب در گره j در ارتباط باشد و معادله ۱-۴، نوع متغیرهای تصمیم را از نوع باینری تعریف می‌کند. جهت خطی‌سازی تابع هدف، توجه به این نکته ضروری است که اگر گره غیر هاب i به یک هاب در گره j تخصیص داده شده باشد، از آنجایی که فقط یک گره هاب داریم لذا تمام گره‌های غیرهاب دیگر نیز به این هاب اختصاص می‌یابند. بنابراین معادله ۱-۱ به‌صورت زیر بازنویسی می‌شود:

$$\begin{aligned}
&\sum_i \sum_j \sum_k W_{ik}(C_{ij} + C_{jk}) Y_{ij} Y_{kj} \\
&= \sum_i \sum_j C_{ij} Y_{ij} (\sum_k W_{ik}) + \sum_j \sum_i C_{ji} Y_{ij} (\sum_k W_{ki}) \\
&= \sum_i \sum_j C_{ij} Y_{ij} (O_i + D_i)
\end{aligned} \tag{۱-۵}$$

[۱] single-HLP

که O_i کل جریان خروجی از گره i و D_i کل جریان ورودی به گره i است. بعد از جایگزینی معادله ۱-۱ با معادله ۱-۵، جواب بهینه مسأله مکانیابی تک- هاب بدست خواهد آمد که در آن ضابطه mini-sum به حداقل می‌رسد. همچنین، تعداد متغیرهای تصمیم از مرتبه $O(n^۲)$ است.

۱-۴-۲ مسأله مکانیابی p - هاب [۱]

در این مسأله، هر گره غیرهاب باید فقط به یک گره هاب تخصیص یابد. از این‌رو، به‌عنوان مسأله مکانیابی p - هاب تک تخصیصی در نظر گرفته می‌شود که اولین بار توسط اُکلی [۸] مطرح گردید.

در این مدل ضابطه mini-sum است، حوزه جواب شبکه است، گره‌های هاب کاملاً به هم متصل‌اند و هر گره غیرهاب تنها با یک گره هاب در ارتباط است. تعداد هاب‌ها برای تعیین مکان، برونی تعریف شده و برابر با p است. حداقل یک و حداکثر دو گره هاب برای جابجایی بین دو گره غیرهاب حائل شده‌اند. بعلاوه، تأسیس هاب‌ها هزینه‌ای در بر ندارد، هاب‌ها بدون ظرفیت‌اند و خروجی‌های مدل باینری هستند. علاوه بر تمام ورودی‌های معرفی شده برای مسأله مکانیابی تک-هاب، در این مسأله عامل تخفیف α معرفی می‌گردد که نشان دهنده‌ی مقیاس اقتصادی برای انتقال جریان بین گره‌های هاب است ($۰ \leqslant \alpha \leqslant ۱$). از آن‌جایی که هزینه حمل و نقل بین دو گره هاب کمتر از هزینه حمل و نقل بین یک گره هاب و گره غیرهاب است، برای محاسبه هزینه جابجایی بین گره‌های هاب، C_{ij} در α ضرب می‌شود. همچنین، خروجی‌های این مدل مشابه مدل قبلی است. بنابراین، ضابطه و محدودیت‌های مدل برای مسأله مکانیابی p - هاب به صورت زیر بیان می‌گردد:

$$\min \quad \sum_i \sum_k C_{ik} Y_{ik} (\sum_j W_{ij}) + \sum_k \sum_i C_{ki} Y_{ik} (\sum_j W_{ji}) + \alpha \sum_i \sum_j \sum_k \sum_m W_{ij} C_{km} Y_{ik} Y_{jm} \qquad (۶-۱)$$

$s.t.$

$$\sum_j Y_{ij} = ۱ \qquad \forall i \qquad (۷-۱)$$

$$\sum_j Y_{jj} = p \qquad (۸-۱)$$

$$Y_{ij} - Y_{jj} \leqslant ۰ \qquad \forall i, j \qquad (۹-۱)$$

$$Y_{ij} \in \{۰, ۱\} \qquad \forall i, j \qquad (۱۰-۱)$$

در فرمولبندی فوق، معادله ۱-۶ کل هزینه حمل و نقل بین گره‌های شبکه را مینیمم می‌کند. عبارت اول این معادله، هزینه اتصال تمام حمل و نقل‌ها از گره غیرهاب i به گره هاب k است وقتی‌که گره i به گره k تخصیص داده شود. عبارت دوم، هزینه اتصال همه‌ی گردش‌های از قبل تعیین شده برای گره غیرهاب i به گره هاب k است به شرط اینکه گره i به گره k اختصاص

[۱] p-HLP

یافته باشد و عبارت سوم، هزینه حمل و نقل جریان‌های بین هابی است. بعلاوه، معادله ۱-۷ تصریح می‌کند که گره غیرهاب i دقیقاً به یک هاب اختصاص داده شده است. معادله ۱-۸ نشان می‌دهد که دقیقاً p گره هاب انتخاب می‌شوند. معادله ۱-۹ تأییدکننده این نکته مهم است که تنها در صورتی‌که هابی در گره j واقع شده باشد آنگاه گره i به گره j تخصیص داده می‌شود. درنهایت، معادله ۱-۱۰ متغیرهای تصمیم را به‌صورت باینری تعریف می‌کند.

۱-۴-۳ مسأله مکانیابی p - هاب میانه [۱]

تابع هدف مدل قبلی غیرخطی بود. بنابراین، کمپبل [۱۹] مدل ذیل را با تابع هدف خطی بیان نمود. در این مدل، هر گره غیرهاب می‌تواند به بیش از یک گره هاب متصل شود. بنابراین، مسأله مکانیابی p - هاب چند تخصیصی نیز نامیده می‌شود. مفروضات این مدل نیز مشابه پیش‌فرض‌های مسأله مکانیابی p - هاب است با این تفاوت که متغیرهای تخصیص، که در اینجا با Z_{ij}^{km} نشان داده می‌شوند، نامنفی فرض شده‌اند ($Z_{ij}^{km} \geqslant ۰$) و گره‌های غیرهاب می‌توانند به چندین گره هاب وصل شوند.

تمام ورودی‌های مسأله مکانیابی p - هاب در اینجا نیز به‌کار برده می‌شوند. بعلاوه، C_{ij}^{km} به‌عنوان هزینه واحد حمل و نقل بین گره آغازی i، گره پایانی j و گره‌های هاب k و m تعریف می‌شود که ترتیب قرارگیری آن‌ها به‌صورت $i \to k \to m \to j$ است.

$$C_{ij}^{km} = C_{ik} + \alpha C_{km} + C_{mj} \qquad (۱۱-۱)$$

خروجی‌های مسأله به‌صورت زیر بیان می‌شوند:

X_j: برابر یک است اگر هابی در گره j واقع شده باشد و در غیر این‌صورت برابر صفر است؛

Z_{ij}^{km}: نشان دهنده‌ی وجود جریان از گره‌ی مبدأ i به مقصد j توسط تسهیلات هاب قرار داده شده در گره‌های k و m است.

بنابراین، فرمول ریاضی مسأله مکانیابی p - هاب میانه چند تخصیصی به‌صورت زیر تعریف می‌شود:

$$\min \quad \sum_i \sum_j \sum_k \sum_m C_{ij}^{km} W_{ij} Z_{ij}^{km} \qquad (۱۲-۱)$$

$$s.t.$$

$$\sum_k X_k = p \qquad (۱۳-۱)$$

$$\sum_k \sum_m Z_{ij}^{km} = ۱ \qquad \forall i, j \qquad (۱۴-۱)$$

[۱] p-Hub median location problem

$$Z_{ij}^{km} \leqslant X_m \qquad \forall i,j,k,m \tag{۱۵-۱}$$

$$Z_{ij}^{km} \leqslant X_k \qquad \forall i,j,k,m \tag{۱۶-۱}$$

$$Z_{ij}^{km} \geqslant ۰ \qquad \forall i,j,k,m \tag{۱۷-۱}$$

$$X_k \in \{۰,۱\} \qquad \forall k \tag{۱۸-۱}$$

با توجه به تابع هدف، معادله ۱-۱۲ کل هزینه حمل و نقل را به حداقل می‌رساند. معادله ۱-۱۳ تضمین می‌کند که دقیقاً p هاب انتخاب می‌شوند. معادله ۱-۱۴ تصریح می‌کند که هر جفت مبدأ و مقصد (i,j) به یک جفت از گره‌های هاب (k,m) تخصیص داده می‌شود. توجه داشته باشید که جفت مبدأ و مقصد (i,j) می‌تواند به یک هاب اختصاص یابد به این‌صورت که اندیس‌های k و m یکسان باشند. معادلات ۱-۱۵ و ۱-۱۶ این اطمینان را بوجود می‌آورند که تقاضا از گره مبدأ i به گره مقصد j نمی‌تواند به جفت (k,m) اختصاص پیدا کند مگر این‌که هر دو گره k و m به‌عنوان گره‌های هاب انتخاب شده باشند. همچنین، معادلات ۱-۱۷ و ۱-۱۸ نوع متغیرهای تصمیم را تعریف می‌کنند.

یکی از مشکلات اصلی فرمولبندی فوق این است که تعداد متغیرهای تخصیص (Z_{ij}^{km}) می‌تواند خیلی بزرگ شود. در واقع، در مسائل مکانیابی هاب شبکه‌ای با این فرض که هر گره مبدأ و مقصد می‌تواند به‌عنوان گره هاب کاندید شود، متغیرهایی از اندازه $O(n^۴)$ وجود خواهندداشت. هنگامیکه تعداد گره‌های شبکه افزایش می‌یابد، اندازه‌ی چنین مدلی سریعاً رشد می‌کند مگراین‌که مجموعه گره‌های هاب کاندید از قبل به‌صورت مدل های مکانیابی گسسته کاهش داده شوند. اگرچه استفاده از فرمولبندی درجه دوم تعداد متغیرهای تصمیم را کاهش می‌دهد اما دستیابی به راه حل سریع مسأله را تضمین نمی‌کند.

۱-۴-۴ مسأله مکانیابی p - هاب با هزینه ثابت پیوند [۱]

کمپبل [۱۸] پیشنهاد کرد که مدل‌های پایه با ذکر هزینه‌ای ثابت برای اتصال گره‌های غیرهاب به گره‌های هاب توسعه داده شوند. چنانکه گره‌های غیرهاب باید به یک گره هاب اختصاص داده شوند، می‌توان هزینه‌ای ثابت برای تمام اتصالات گره‌های غیرهاب به گره‌های هاب در نظر گرفت. ورودی‌ها و خروجی‌های این مدل مشابه مسأله مکانیابی p - هاب میانه هستند. بعلاوه، g_{ik} هزینه ثابت اتصال گره غیرهاب i به هاب قرار داده شده در گره k است. علاوه براین، W_{ik} متغیری باینری است که اگر برابر یک باشد بیانگر انتخاب لینک (i,k) خواهدبود. تابع هدف این مدل مشابه تابع هدف مسأله مکانیابی p - هاب میانه است با این تفاوت که انتصاب هزینه زیر به ضابطه موردنظر اضافه می‌شود:

$$\sum_i \sum_k g_{ik} W_{ik} \tag{۱۹-۱}$$

[۱] p-HLP with fixed link cost

۱-۴-۵ مسأله مکانیابی p - هاب با حداقل مقدار جریان روی پیوندها [۱]

مسأله‌ای شبیه به مسأله مکانیابی p - هاب میانه تک تخصیصی، توسط کمپبل [۱۸] مطرح شد. او اظهار داشت که گاهی اوقات به جای تمرکز بر روی این مطلب که هر گره غیرهاب باید به یک گره هاب تخصیص داده شود، کاربردی‌تر است تا جنبه‌ی خاصی برای این امر که جریان بین هر گره غیرهاب و گره هاب بزرگتر یا مساوی با حداقل مقدار آستانه‌ای جریان باشد قائل شویم. مدل موردنظر متشابهاً به‌صورت مسأله مکانیابی p - هاب با درنظرگرفتن حداقل مقدار آستانه برای جریان روی هر پیوند غیرهاب به هاب، فرمولبندی می‌شود. علاوه بر تمام ورودی‌های مسأله مکانیابی p - هاب که در اینجا نیز استفاده می‌شوند پارامتر ورودی زیر معرفی می‌گردد:

L_{ik}: حداقل مقدار آستانه برای جریان بین گره غیرهاب i و گره هاب k.

خروجی‌های مدل مشابه خروجی‌های مسأله مکانیابی p - هاب میانه هستند. همچنین، معادلات زیر بایستی به محدودیت‌های مسأله مکانیابی p - هاب میانه اضافه شوند:

$$Y_{ik} + Y_{jm} - ۲Z_{ij}^{km} \geqslant ۰ \qquad \forall i,j,k,m \tag{۱-۲۰}$$

$$\sum_{m}\sum_{j} W_{ij} Z_{ij}^{km} + \sum_{p}\sum_{s} W_{pi} Z_{pi}^{sk} \geqslant L_{ik} Y_{ik} \qquad \forall i,k \tag{۱-۲۱}$$

در معادله ۱-۲۰ تقاضا از مبدأ i به مقصد j می‌تواند فقط توسط هاب‌های موجود در گره‌های k و m تعیین مسیر شود، اگر

- گره غیرهاب i به هاب قرارداده شده در گره k تخصیص داده شده باشد؛
- گره غیرهاب j به هابی که در گره m قرار دارد تخصیص داده شده باشد.

از طرفی معادله ۱-۲۱ وجود حداقل مقدار آستانه‌ای جریان روی لینک‌های اتصال گره غیرهاب به گره هاب را تضمین می‌کند. عبارت اول این معادله، کل جریان از گره غیرهاب i به گره هاب k و از آن‌جا به هر جفت گره هاب-مقصد و عبارت دوم، کل جریان از هر جفت گره مبدأ-هاب به گره هاب k و از آنجا به مقصد i را محاسبه می‌کند. مجموع این دو عبارت، کل جریان بین دو گره i و k است که باید بیشتر از آستانه مشخص شده باشد با این فرض که لینک اتصال گره غیرهاب به گره هاب ایجاد شده است .

۱-۴-۶ مسأله مکانیابی p - هاب با ظرفیت محدود [۲]

کمپبل [۱۸] مسأله‌ای را مطرح نمود که در آن ظرفیت تسهیلات هاب، جریان‌های شبکه را محدود می‌کرد. در واقع، محدودیت ظرفیت گره هاب بدان معنی است که کل جریان ورودی یا خروجی این گره باید کمتر از یا برابر با یک مقدار ثابت باشد. این مسأله مشابه مسأله مکانیابی p - هاب میانه و با همان ورودی‌ها و خروجی‌ها فرمولبندی می‌شود. بعلاوه،

[۱] p-Hub with Minimum-value flow on links [۲] p-HLP with limited capacity

θ_k نشان دهنده ظرفیت هاب واقع در گره k است. محدودیت زیر باید همراه ضابطه اصلی و محدودیت‌های مسأله مکانیابی p - هاب میانه در نظر گرفته شود:

$$\sum_m \sum_i \sum_j W_{ij} Z_{ij}^{km} + \sum_s \sum_i \sum_j W_{ij} Z_{ij}^{sk} \leqslant \theta_k X_k \qquad \forall k \qquad (۲۲-۱)$$

در این محدودیت، عبارت‌های اول و دوم به ترتیب بیانگر جریان‌های ورودی و خروجی گره هاب موردنظر هستند.

۱-۴-۷ مسأله مکانیابی p - هاب پیوسته [۱]

معمولاً مسائل مکانیابی هاب یا به‌صورت شبکه‌ای و یا گسسته مدل‌سازی می‌شوند. با این حال، مطالعاتی در زمینه مسائلی با حوزه جواب پیوسته انجام گرفته است. در مسائل مکانیابی هاب پیوسته، دامنه گره‌های هاب مجموعه‌ای از گره‌های بالقوه روی یک گراف نیست بلکه این مجموعه روی یک صفحه و یا روی یک کره قرار دارد. در این مسأله به همان گونه‌ای که در مسأله مکانیابی p - هاب تک تخصیصی اشاره شد هر گره غیرهاب فقط به یک گره هاب اختصاص می‌یابد . حالات خاص این مسأله که شامل تنها یک و یا دو گره هاب است اولین بار توسط اُکلی مد نظر قرار گرفت. بعد از آن، محققان دیگری این مدل را با در نظرگرفتن p گره هاب گسترش دادند.

در این مدل ضابطه mini-sum است، حوزه جواب صفحه‌ای پیوسته است، گره‌های هاب کاملاً با هم پیوند دارند و هر گره غیرهاب تنها به یک گره هاب متصل است، تعداد گره‌های هاب که قرار است مکانیابی شوند به طور برونی تعریف شده و حداقل یک یا حداکثر دو گره هاب باید برای جابجایی بین دو گره غیرهاب قرار گیرند. علاوه براین، هزینه ثابت تأسیس هاب در نظر گرفته نمی‌شود، هاب‌ها بدون ظرفیت‌اند و متغیرهای تصمیم باینری هستند.

ورودی‌های مسأله شبیه به ورودی‌های مسأله مکانیابی p - هاب هستند. ضمن اینکه N_i بردار موقعیت مکانی گره غیرهاب i است. بعلاوه، خروجی‌های مدل مشابه خروجی‌های مسأله مکانیابی p - هاب می‌باشند و متغیر تصمیم P_k بردار مکان گره هاب k را نشان می‌دهد $(k = ۱, \cdots, p)$. با درنظرگرفتن $d(a,b)$ به‌عنوان فاصله اقلیدسی بین دو گره a و b، ضابطه و محدودیت‌های مدل به‌صورت زیر بیان می‌شوند:

$$\min \quad \sum_i \sum_j \sum_k \sum_m W_{ij} Y_{ik} Y_{jm} (d(N_i, P_k) + \alpha d(P_k, P_m) + d(N_j, P_m)) \qquad (۲۳-۱)$$

$s.t.$

$$\sum_j Y_{ij} = ۱ \qquad \forall i \qquad (۲۴-۱)$$

$$P_k = (a_k, b_k) \qquad k = ۱, \cdots, p \qquad (۲۵-۱)$$

$$Y_{ij} \in \{۰, ۱\} \qquad \forall i, j \qquad (۲۶-۱)$$

[۱] continuous p-HLP

در معادله ۱-۲۳ هزینه کل حمل‌ونقل به حداقل می‌رسد. عبارت اول، هزینه اتصال تمام نقل و انتقالات شروع شده از گره‌های غیرهاب به سمت گره‌های هاب مربوطه‌شان را مورد محاسبه قرار می‌دهد. عبارت دوم ، هزینه حمل و نقل بین گره‌های هاب است در حالیکه عبارت سوم بیانگر هزینه حمل و نقل همه‌ی جابجایی‌ها بین گره‌های هاب به سوی گره غیرهاب می‌باشد. معادله ۱-۲۴ تصریح می‌کند که هر گره غیرهاب دقیقاً به یک گره هاب تخصیص داده می‌شود. نهایتاً، معادلات ۱-۲۵ و ۱-۲۶ نوع متغیرهای تصمیم را تعریف می‌کنند. در این مدل، تعامل بین گره‌های هاب توسط گره‌های غیرهاب تخصیص داده شده به آن‌ها تعیین می‌شود.

۱-۴-۸ مسأله مکانیابی p - هاب چندهدفه [۱]

کوستا [۲] ، کاپتیوو [۳] و کلیماکو [۴] [۲۰] مسأله مکانیابی هاب چند هدفه را پیشنهاد کردند. در این مسأله، اولین تابع هدف وظیفه به حداقل رساندن هزینه کل حمل و نقل را بر عهده دارد در حالیکه دومین تابع هدف به دنبال این است تا حداکثر زمانی که گره‌های هاب برای پردازش جریان لازم دارند و یا به‌عبارت دیگر حداکثر زمان سرویس گره‌های هاب را مینیمم کند. در این مدل، ضابطه mini-sum و mini-max است، حوزه جواب شبکه است، گره‌های هاب به‌طور کامل با یکدیگر در ارتباط هستند و هر گره غیرهاب تنها به یک گره هاب تخصیص می‌یابد. تعداد هاب‌ها برای تعیین مکان برونی‌اند و باید حداقل یک یا حداکثر دو گره هاب برای جابجایی بین دو گره غیرهاب استفاده شود. علاوه‌براین، هزینه‌های ثابت برای شروع سرویس در گره‌های هاب در نظر گرفته نمی‌شوند، هاب‌ها بدون ظرفیت‌اند و متغیرهای تصمیم باینری هستند.

ورودی‌ها و خروجی‌های این مسأله مشابه مسأله مکانیابی p - هاب است. بعلاوه، متغیر ورودی T_k معرفی می‌گردد که بیانگر مدت زمانی است که گره هاب k برای پردازش یک واحد جریان نیاز دارد. ضابطه و محدودیت‌های مدل به شرح زیر است:

$$\min \quad \sum_i \sum_j \sum_k \sum_m W_{ij} Y_{ik} Y_{jm} (C_{ik} + \alpha C_{km} + C_{jm}) \qquad (۱-۲۷)$$

$$\min \quad \max_k \left\{ T_k \sum_i \sum_j W_{ij} Y_{ik} + \sum_i \sum_j W_{ji} Y_{jm} Y_{ik} \right\} \qquad (۱-۲۸)$$

$s.t.$

$$\sum_k Y_{ik} = ۱ \qquad \forall i \qquad (۱-۲۹)$$

$$\sum_k Y_{kk} = p \qquad (۱-۳۰)$$

$$Y_{ik} - Y_{kk} \leqslant ۰ \quad \forall i, k \qquad (۱-۳۱)$$

$$Y_{ik} \in \{۰, ۱\} \qquad \forall i, k \qquad (۱-۳۲)$$

---

[۱] multi-objective p-HLP [۲] Costa [۳] Captivo [۴] Climaco

نقش معادله ۱-۲۷ مینیمم کردن کل هزینه حمل‌ونقل در شبکه هاب است. معادله ۱-۲۸ حداکثر زمان سرویسی را که گره هاب k برای پردازش کل جریان ورودی و خروجی صرف می‌کند به حداقل می‌رساند. معادله ۱-۲۹ تصریح می‌کند که هر گره غیرهاب i باید دقیقاً به یک گره هاب تخصیص داده شود. معادله ۱-۳۰ تأکید کننده انتخاب p گره هاب است. معادله ۱-۳۱ تضمین می‌کند که هیچ گره غیرهابی نمی‌تواند با یک گره هاب در ارتباط باشد تا وقتی‌که آن گره هاب انتخاب شود. همچنین، معادله ۱-۳۲ نوع متغیرهای تصمیم را از نوع باینری تعریف می‌کند.

۱-۴-۹ مسأله مکانیابی p - هاب مرکزی [۱]

مسأله مکانیابی p - مرکز از مهم‌ترین تغییرات مسائل مکانیابی تسهیلات است. این مسأله برای مسائل مکانیابی تسهیلات اضطراری مفید است. مسأله مکانیابی p - هاب مرکزی مشابه مسأله مکانیابی p - مرکز است با این فرض که یک سری از جفت گره‌های مبدأ و مقصد در مسأله مکانیابی هاب وجود دارند که می‌توانند به نمایندگی از گره‌های تقاضا در مسأله p - مرکز در نظر گرفته شوند. مسأله مکانیابی p - هاب مرکزی براساس ضابطه Mini-Max تعریف می‌شود که در آن حداکثر هزینه جفت‌های مبدأ و مقصد مینیمم می‌گردد. این مسأله برای موقعیت‌هایی نظیر جابجایی کالاهای فاسدشدنی در شبکه‌های حمل‌ونقل قابل اجراست که در اصل توسط کمپبل [۱۸] مطرح شد. نمادهای این مسأله مشابه مسأله مکانیابی p - هاب میانه است با این تفاوت که ضابطه در این‌جا Mini-Max است. ضابطه و محدودیت‌های مدل به شرح زیر هستند:

$$\min \quad \max_{i,j,k,m} \left\{ C_{ij}^{km} W_{ij} Z_{ij}^{km} \right\} \tag{۱-۳۳}$$

$$s.t.$$

$$\sum_{k} X_k = p \tag{۱-۳۴}$$

$$\sum_{k} \sum_{m} Z_{ij}^{km} = ۱ \qquad \forall i, j \tag{۱-۳۵}$$

$$Z_{ij}^{km} \leqslant X_k \quad \forall i, j, k, m \tag{۱-۳۶}$$

$$Z_{ij}^{km} \leqslant X_m \quad \forall i, j, k, m \tag{۱-۳۷}$$

$$Z_{ij}^{km} \geqslant ۰ \quad \forall i, j, k, m \tag{۱-۳۸}$$

$$X_k \in \{۰, ۱\} \qquad \forall k \tag{۱-۳۹}$$

در معادله ۱-۳۳ ماکزیمم هزینه حمل‌ونقل از گره‌های مبدأ به گره‌های مقصد به حداقل می‌رسد. همچنین، برای معادلات ۱-۳۴ تا ۱-۳۹ توضیحاتی مشابه محدودیت‌های مسأله مکانیابی p - هاب میانه بیان می‌شود.

[۱] p-Hub center location problem

۱-۴-۱۰ مسأله مکانیابی p - هاب پوششی [۱]

مسائل مکانیابی p - هاب پوششی تعمیمی از مسائل مکانیابی پوششی کلاسیک هستند که طرح اولیه آن توسط کمپبل [۱۸] پیشنهاد شد. در این نوع مسائل، تسهیلات هاب به نحوی تعیین مکان می‌شوند که جفت گره مبدأ و مقصد متشکل از دو گره غیرهاب توسط یک جفت از گره‌های هاب پوشانده شود. در واقع، تنها در صورتی جفت‌های مبدأ و مقصد تحت پوشش هستند که تسهیلات هابی در فواصل از پیش تعیین شده از لینک‌هایشان وجود داشته باشند. به همین دلیل، هزینه حمل‌ونقل از گره‌های آغازی به گره‌های پایانی از طریق گره‌های هاب انتخاب شده باید بزرگتر و یا مساوی با یک مقدار از پیش تعیین شده باشد.

$$C_{ij}^{km} \leqslant \gamma_{ij} \qquad (۴۰-۱)$$

این مسأله به دو مسأله دیگر یعنی مسأله مکانیابی هاب مجموعه پوشش و مسأله مکانیابی هاب حداکثر پوشش توسعه داده می‌شود که در ادامه مورد بحث قرار خواهند گرفت.

۱-۴-۱۰-۱ مسأله مکانیابی هاب مجموعه پوشش [۲]

این مسأله توسیع خاصی از مسأله مکانیابی هاب پوششی است که مشابه مسأله مکانیابی p - هاب میانه فرمول‌بندی می‌شود با این تفاوت که قبل از حل مسأله از تعدا گره‌های هابی که باید تعیین مکان شوند اطلاعی در دست نیست. از این‌رو، هزینه‌ی ثابت تأسیس تسهیلات هاب در نظر گرفته شده‌است. ورودی‌های مدل عبارتند از:

F_k: هزینه ثابت ایجاد یک هاب در گره k

C_{ij}^{km}: هزینه انتقال از گره مبدأ i به گره مقصد j توسط گره‌های هاب قرار داده شده در گره‌های k و m

γ_{ij}: حداکثر هزینه برای پوشش لینک‌های اتصال گره‌های تقاضای i به j

V_{ij}^{km}: برابر با یک است اگر هاب‌های قرار داده شده در گره‌های k و m بتوانند جفت گره (i,j) را بپوشانند

خروجی‌های مدل شبیه به خروجی‌های مسأله مکانیابی p - هاب میانه‌اند. همچنین، ضابطه و محدودیت‌های مدل موردنظر به‌صورت زیر هستند:

$$\min \quad \sum_k F_k X_k \qquad (۴۱-۱)$$

$$s.t.$$

$$\sum_k \sum_m V_{ij}^{km} Z_{ij}^{km} \geqslant ۱ \qquad \forall i,j \qquad (۴۲-۱)$$

[۱] p-Hub covering location problem [۲] Hub set covering location problem

$$Z_{ij}^{km} \leqslant X_k \quad \forall i,j,k,m \tag{۴۳-۱}$$

$$Z_{ij}^{km} \leqslant X_m \quad \forall i,j,k,m \tag{۴۴-۱}$$

$$Z_{ij}^{km} \geqslant ۰ \quad \forall i,j,k,m \tag{۴۵-۱}$$

$$X_k \in \{۰,۱\} \qquad \forall k \tag{۴۶-۱}$$

معادله ۱-۴۱ هزینه کل افتتاح تسهیلات هاب جدید را به حداقل می‌رساند و معادله ۱-۴۲ تصریح می‌کند که هر جفت تقاضا حداقل یک بار توسط یک جفت هاب تحت پوشش قرار می‌گیرد. محدودیت‌های دیگر مدل مشابه محدودیت‌های مسائل قبلی قابل توضیح است.

۱-۴-۱۰-۲ مسأله مکانیابی p - هاب حداکثر پوشش [۱]

فرمول پیشنهادی برای این مسأله شبیه به مسأله مکانیابی p - هاب میانه است. تعداد گره‌های هاب به‌طور برونی مشخص می‌شود و هزینه ثابت تأسیس تسهیلات هاب لحاظ نمی‌گردد. تابع هدف و محدودیت‌های این مسأله عبارتند از:

$$\max \quad \sum_i \sum_j \sum_k \sum_m W_{ij} V_{ij}^{km} Z_{ij}^{km} \tag{۴۷-۱}$$

$s.t.$

$$\sum_k X_k = p \qquad \forall i,j \tag{۴۸-۱}$$

$$\sum_k \sum_m Z_{ij}^{km} = ۱ \qquad \forall i,j \tag{۴۹-۱}$$

$$Z_{ij}^{km} \leqslant X_k \quad \forall i,j,k,m \tag{۵۰-۱}$$

$$Z_{ij}^{km} \leqslant X_m \quad \forall i,j,k,m \tag{۵۱-۱}$$

$$Z_{ij}^{km} \geqslant ۰ \quad \forall i,j,k,m \tag{۵۲-۱}$$

$$X_k \in \{۰,۱\} \qquad \forall k \tag{۵۳-۱}$$

معادله ۱-۴۷ مقدار تقاضای حمل‌ونقل تحت پوشش را بیشینه می‌کند. محدودیت‌های دیگر همانند مسأله مکانیابی p -هاب میانه قابل توضیح است.

ادبیات مسائل مکانیابی هاب نشان می‌دهد که مسائل مکانیابی هاب میانه در مقایسه با مسائل مکانیابی هاب پوششی و مرکزی توجه بیشتری را به خود جلب کرده‌اند. در حقیقت، بیشتر مطالعات در زمینه مسائل پوششی و مرکزی در چند سال اخیر منتشر شده‌اند.

[۱] p-Hub maximal covering location problem

۱-۵ فرمولبندی مسأله مکانیابی هاب تک تخصیصی بدون ظرفیت

در این بخش، به بیان یک مدل از مسائل مکانیابی هاب به نام مسأله مکانیابی هاب تک تخصیصی بدون ظرفیت (USAHLP)[۱] می‌پردازیم که محوریت بحث در فصل‌های بعدی، پیرامون این مدل خواهد بود. بیشتر مسائل مکانیابی هاب به‌عنوان مسائل NP-hard شناخته شده‌اند. مسأله USAHLP نیز در این کلاس از پیچیدگی واقع شده‌است. در این مدل، هر گره غیرهاب تنها به یک گره هاب تخصیص می‌یابد، محدودیت ظرفیتی بر روی گره‌های هاب وجود ندارد و برای تأسیس هر گره هاب، هزینه‌ای ثابت در نظر گرفته می‌شود. هدف USAHLP عبارت است از تعیین تعداد هاب‌ها، تعیین مکان هاب‌ها و تخصیص گره‌های غیرهاب به هاب‌ها به‌گونه‌ای که مجموع هزینه‌های حمل و نقل جریان و تأسیس هاب‌ها مینیمم گردد.

مدل ریاضی مسأله USAHLP به‌صورت زیر بیان می‌شود [۲۱]. مجموعه I از n گره مشخص در یک شبکه را در نظر بگیرید. فرض کنید C_{ij} هزینه حمل‌ونقل یک واحد جریان بین گره‌های i و j باشد. این ماتریس دارای وجودی انتزاعی است و در کاربردها می‌تواند اندازه‌های مختلفی را نمایش دهد که اغلب نسبی است مگر این‌که به فاصله بین i و j مربوط باشد. بنابراین، اصطلاح " نزدیک‌ترین گره به گره k " برای گره‌ای با کمترین هزینه حمل‌ونقل از گره k استعمال می‌گردد.

همچنین، W_{ij} ماتریس جریان است که اعضای آن مقدار جریان از مبدأ i به مقصد j را بیان می‌کنند که در حالت کلی، نامساوی‌های $W_{ij} \neq W_{ji}$ و $W_{ii} \neq ۰$ برقرارند. فرض کنید که $H \subseteq I$ مجموعه هاب‌های تأسیس شده باشد. از آن‌جایی که حمل‌ونقل مستقیم بین گره‌های غیرهاب مجاز نیست، کل جریان از گره مبدأ i به گره مقصد j باید به ترتیب توسط گره‌های k و l تعیین مسیر شوند که $k, l \in H$. توجه داشته باشید که ممکن است گره‌های k و l یکسان باشند که در این‌صورت انتقال جریان توسط یک هاب صورت می‌گیرد. پارامترهای χ و α و δ به ترتیب منعکس کننده نرخ‌های واحد هزینه‌ها برای گردآوری(مبدأ-هاب)، واگذاری(هاب-هاب) و توزیع(هاب-مقصد) هستند. از این‌رو، هزینه انتقال یک واحد جریان از مبدأ i به مقصد j توسط هاب‌های k و l به صورت زیر محاسبه می‌شود:

$$\chi C_{ik} + \alpha C_{kl} + \delta C_{lj}$$

به دلیل کارآمدی‌های انتقال درون-هابی، معمولاً رابطه‌ی $\chi, \delta > \alpha$ برقرار است. در ادبیات، این پارامترها معمولاً به‌صورت $\alpha < ۱$ و $\chi, \delta = ۱$ در نظر گرفته می‌شوند.

بعلاوه، ثابت f_k هزینه ثابت تأسیس یک هاب در گره k است.متغیرهای تصمیم باینری Z_{ij} در صورت تخصیص گره i به گره هاب j مقدار ۱ و در غیر این‌صورت مقدار ۰ را اتخاذ می‌کنند. با تعریف $Z_{kk} = ۱$ بدین معنی که k گره‌ای هاب است داریم: $Z_{kk} = ۱ \Leftrightarrow k \in H$. برای بیان مقدار جریان شروع شده از گره i، گردآوری شده در گره هاب k و توزیع شده توسط هاب l، متغیرهای نامنفی Y^i_{kl} معرفی می‌شوند که عدم وجود جریان مستقیم بین گره‌های غیرهاب i و j از علل معرفی این دسته از متغیرهاست. با توجه به این‌که هزینه‌های حمل‌ونقل بین گره‌ها دارای خاصیت نامساوی مثلثی است برای جابجایی جریان، حداکثر دو هاب مورد استفاده قرار می‌گیرند.

[۱] Uncapacitated Single Allocation Hub Location Problem

جهت ساده‌سازی نمادها، مقدار جریان خروجی از گره i با O_i و مقدار جریان وارد شده به گره j با D_j نمایش داده می‌شوند. بعبارتی:

$$O_i = \sum_{j \in I} W_{ij} \quad , \quad D_j = \sum_{i \in I} W_{ij}$$

با استفاده از نمادهایی که در بالا اشاره شد، مسأله USAHLP به‌صورت زیر فرمولبندی می‌شود:

$$\min \quad \sum_{i \in I} \sum_{k \in I} (C_{ik} + Z_{ik}(\chi O_i + \delta D_i) + \sum_{i \in I} \sum_{k \in I} \sum_{l \in I} \alpha C_{kl} Y_{kl}^i + \sum_{k \in I} f_k Z_{kk} \qquad (۵۴-۱)$$

$s.t.$

$$\sum_{k \in I} Z_{ik} = ۱ \qquad \forall i \in I \qquad (۵۵-۱)$$

$$Z_{ik} \leqslant Z_{kk} \qquad \forall i, k \in I \qquad (۵۶-۱)$$

$$\sum_{l \in I} Y_{kl}^i - \sum_{l \in I} Y_{lk}^i = O_i Z_{ik} - \sum_{j \in I} W_{ij} Z_{jk} \qquad \forall i, k \in I \qquad (۵۷-۱)$$

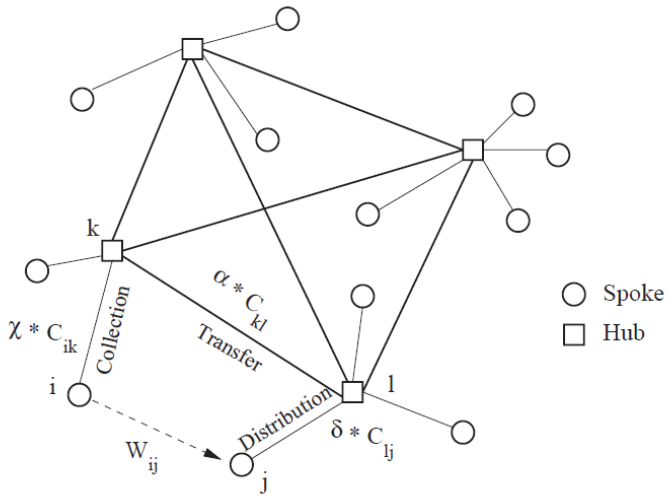
$$Y_{kl}^i \geqslant ۰ \qquad \forall i, k, l \in I \qquad (۵۸-۱)$$

$$Z_{ik} \in \{۰, ۱\} \qquad \forall i, k \in I \qquad (۵۹-۱)$$

در فرمولبندی فوق، مجموع هزینه‌های حمل‌ونقل بین تمامی جفت‌های مبدأ و مقصد که توسط تسهیلات هاب صورت می‌گیرند و هزینه‌های ثابت مربوط به تعیین مکان مجموعه هاب‌ها بواسطه تابع هدف ۱-۵۴ به حداقل می‌رسد. مجموعه محدودیت‌های ۱-۵۵ تضمین می‌کنند که هر گره دقیقاً به یک گره هاب تخصیص می‌یابد. جهت حصول اطمینان از عدم وجود حمل‌ونقل مستقیم بین گره‌های غیرهاب، محدودیت‌های ۱-۵۶ معرفی می‌گردند که تصریح کننده امکان تخصیص گره‌های غیرهاب فقط به گره‌های هاب و همچنین تخصیص هر گره هاب به خودش می‌باشند. برقراری برابری جریان در شبکه با محدودیت‌های ۱-۵۷ بیان می‌شود. نهایتاً، محدودیت‌های ۱-۵۸ و ۱-۵۹ به ترتیب نوع متغیرهای Y_{kl}^i و Z_{ij} را به صورت نامنفی و باینری تعریف می‌کنند. در شکل ۱-۵ نمونه‌ای از یک شبکه هاب تک تخصیصی با چهار گره هاب نشان داده شده است.

۱-۶ رویکردهای حل مسأله

تمام روش‌هایی که در ادبیات مسائل مکانیابی هاب برای این مسائل مطرح شده‌اند، حتی در مواردی که روش‌های پیشنهاد شده متعلق به کلاس یکسانی باشند، مسأله خاص به حساب می‌آیند. بنابراین، دور از ذهن نیست که با مسائلی مواجه شویم که با نام یا فرمولبندی مشابه از نظر ساختار متفاوت باشند که در نتیجه رویکرد متفاوتی برای حل آن‌ها موردنیاز است [۲۱].



شکل ۱-۵: نمایی کامل از شبکه هاب تک تخصیصی

رابطه بین تصمیمات مربوط به تعیین محل هاب‌ها و طراحی شبکه، باعث دشواری حل مسائل مکانیابی هاب شده‌است. بنابراین، در طی دو دهه گذشته تلاش‌های قابل توجهی جهت توسعه الگوریتم‌هایی انجام گرفته است که قادر به یافتن جواب‌های با کیفیت بالا برای کلاس‌های مختلف این مسائل، به ویژه موارد واقع گرایانه‌تر یعنی نمونه‌های بزرگ مقیاس، باشند. برخی از این الگوریتم‌ها قادر به ارزیابی کیفیت جواب‌های به‌دست آمده می‌باشند و برخی از آن‌ها می‌توانند ثابت کنند که جواب به‌دست آمده بهینه است [۳].

الگوریتم‌های دقیق برای حل نمونه‌های کوچکتر مسائل مکانیابی هاب و الگوریتم‌های ابتکاری یا فراابتکاری برای نمونه‌های بزرگتر این مسائل به‌کار برده می‌شوند. به عنوان یک واقعیت، اگر چه نمونه‌های بزرگی هستند که می‌توانند با روش‌های دقیق مورد رسیدگی قرار گیرند، با این حال توسعه رویکردهای فراابتکاری به بسیاری از برنامه های دنیای واقعی کمک زیادی کرده است که در آن‌ها جواب‌های بهینه یا نزدیک به بهینه در زمان محاسباتی کمتری بدست می‌آیند [۱۶].

فصل ۲

کاربرد الگوریتم جستجوی ممنوعه و الگوریتم ژنتیک در حل مسائل مکانیابی هاب

۲-۱ مقدمه‌ای بر الگوریتم‌های بهینه‌سازی فراابتکاری

در این بخش، سعی شده‌است نگاهی کلی به الگوریتم‌های فراابتکاری [۱] و تقسیم‌بندی‌های آن‌ها ایجاد شود [۱]. انواع مسائل بهینه‌سازی [۲] ، شامل بهینه‌سازی گسسته و پیوسته، همچنین پیچیدگی محاسباتی الگوریتم‌ها و پیچیدگی محاسباتی مسائل، برای نشان دادن ضرورت استفاده از الگوریتم‌های فراابتکاری تشریح شده‌است. روش‌های حل مسائل بهینه‌سازی شامل روش‌های دقیق، ابتکاری و فراابتکاری تعریف شده‌اند. در انتهای این بخش، تقسیم‌بندی الگوریتم‌های مبتنی بر یک جواب و مبتنی بر جمعیت ارائه شده‌است.

۲-۱-۱ مسائل بهینه‌سازی

در اکثر زمینه‌های تحقیقاتی می‌توان مسئله موردنظر را به‌صورت یک مسئله بهینه‌سازی بیان کرد. هدف از مسائل بهینه‌سازی، پیدا کردن جواب بهینه است. در این نوع از مسائل ، یک یا چند تابع هدف خطی یا غیرخطی تعریف می‌شود که باید در عین حال که تمامی پارامترها مورد توجه قرار گیرند، کمینه یا بیشینه شود. معمولاً در مسائل بهینه‌سازی محدودیت‌هایی نیز تعریف می‌شوند که ممکن است شامل نامعادلات خطی یا غیرخطی و یا اجبار گسسته بودن متغیرهای تصمیم باشند. تمامی جواب‌های موجود باید در این محدودیت‌ها صدق کنند، در غیر این‌صورت جواب‌ها موجه نخواهند بود. مسائل بهینه‌سازی، از نظر نوع متغیرهای تصمیم می‌تواند به دو گروه مسائل گسسته که مسئله یافتن جواب‌های بهینه در این دسته از مسائل،

[۱] metaheuristic algorithms [۲] optimization problems

بهینه‌سازی ترکیبی [1] نامیده می‌شود و مسائلی با متغیرهای پیوسته تقسیم‌بندی شود.

۲-۱-۲ پیچیدگی محاسباتی

یک الگوریتم برای حل مسئله، نیاز به دو منبع مهم دارد: زمان و فضای حافظه.

تعریف ۲-۱-۱. پیچیدگی زمانی یک الگوریتم، برابر با تعداد گام‌های مورد نیاز برای حل یک مسئله با اندازهٔ n است که n بیانگر ابعاد مسئله است.

پیچیدگی معمولاً براساس بدترین حالت تعریف می‌شود. هدف از تعیین پیچیدگی محاسباتی یک الگوریتم، یافتن تعداد دقیق گام‌های الگوریتم نیست؛ بلکه هدف، یافتن یک حد بالا برای تعداد گام‌هاست. نماد O بزرگ، یکی از نمادهای رایج در تحلیل پیچیدگی الگوریتم‌ها است. بنابر تعریف، یک الگوریتم دارای پیچیدگی $f(n) = O(g(n))$ است، اگر ثابت‌های مثبت n_0 و c وجود داشته باشند، به‌طوریکه به ازای هر $n > n_0$ نامساوی $f(n) \leq c.g(n)$ برقرار باشد. در این حالت، $g(n)$ حد بالای $f(n)$ است. این نماد می‌تواند برای محاسبه‌ی پیچیدگی زمانی یا فضای حافظه‌ای یک الگوریتم مورد استفاده قرار گیرد. در بحث پیچیدگی محاسباتی الگوریتم‌ها، می‌توان آن‌ها را در یکی از دو گروه زیر در نظر گرفت:

- الگوریتم با زمان چندجمله‌ای؛ پیچیدگی این الگوریتم $O(p(n))$ است که در آن $p(n)$ یک تابع چندجمله‌ای از n می‌باشد. یک تابع چندجمله‌ای با درجه k، می‌تواند به‌صورت زیر تعریف شود:

$$p(n) = a_k n^k + \cdots + a_j n^j + \cdots + a_1 n + a_0$$

  که در آن $a_k > 0$ و $a_j > 0$ ، $1 \leq j \leq k-1$. در نتیجه، این الگوریتم دارای پیچیدگی چندجمله‌ای $O(n^k)$ است.

- الگوریتم با زمان نمایی؛ پیچیدگی این الگوریتم $O(c^n)$ است که در آن c یک عدد حقیقی ثابت بزرگتر از یک می‌باشد.

توجه داشته باشید که زمان جستجوی یک الگوریتم با توجه به اندازه‌ی مسئله افزایش می‌یابد. مقایسه زمان محاسباتی الگوریتم‌ها، با توجه به پیچیدگی محاسباتی الگوریتم و اندازه مسئله، زمان محاسباتی بسیار بالای پیچیدگی نمایی را در مقابل چندجمله‌ای نشان می‌دهد.

تعریف ۲-۱-۲. پیچیدگی محاسباتی یک مسئله، برابر است با پیچیدگی محاسباتی بهترین الگوریتمی که آن را حل می‌کند.

[1] combinatorial optimization

نظریه‌ی پیچیدگی مسائل، براساس مسائل تصمیم [1] تعریف می‌شود. یک مسئله تصمیم، همیشه یک جواب "بلی" یا "خیر" دارد. هر مسئله بهینه‌سازی، قابل تبدیل به یک مسئله تصمیم است. از لحاظ پیچیدگی محاسباتی، مسائل تصمیم به دو کلاس مهم P و NP تقسیم می‌شوند.

تعریف ۲-۱-۳. کلاس P [2]

کلاس پیچیدگی P، شامل مجموعه‌ای از مسائل تصمیم است که الگوریتم‌های قطعی برای پیداکردن جواب آن‌ها وجود دارد. به‌عبارت دیگر، تمامی مسئله‌های تصمصم که دارای جوابی از مرتبه چندجمله‌ای باشند در این کلاس قرار دارند. تعریف دیگری که از مسأله‌های کلاس P ارائه می‌شود این است که هرگاه الگوریتمی وجود داشته باشد که بتواند به ازای هر ورودی به طول n در حداکثر cn^k مرحله (k و c معرف اعداد ثابت مستقل از n) جواب درست بدهد، آنگاه گوییم مسأله می‌تواند در زمان چندجمله‌ای $O(n^k)$ حل شود و آن‌را در کلاس P قرار می‌دهیم.

تعریف ۲-۱-۴. کلاس NP [3]

کلاس پیچیدگیNP، شامل مجموعه‌ای از مسائل تصمیم است که می‌توان آن‌ها را توسط الگوریتم‌های غیرقطعی چندجمله‌ای حل کرد. الگوریتم‌های غیرقطعی چندجمله‌ای، شامل الگوریتم‌های دو مرحله‌ای است. در مرحله اول، یک جواب کاندید به‌صورت تصادفی تولید می‌شود. در مرحله دوم، صحه‌گذاری جواب کاندید انجام می‌شود. در صورتیکه جواب کاندید بیان کننده‌ی یک جواب مسأله باشد، مقدار "بله" و در غیر این‌صورت مقدار "خیر" به‌عنوان جواب نهایی مسأله تصمیم برگردانده می‌شود. بنابراین، غیرقطعی بودن الگوریتم مربوط به مرحله‌ی اول و زمان چندجمله‌ای مربوط به مرحله‌ی دوم الگوریتم‌هاست. در حالت کلی، تمامی مسائلی که جواب آن‌ها توسط الگوریتم با زمان چندجمله‌ای قابل بررسی باشد مسائل NP هستند. بنابراین، هر مسأله P یک مسأله NP خواهد بود.

مسائل NP به دو دسته مسائل NP- کامل و NP- سخت تقسیم می‌شوند:

تعریف ۲-۱-۵. کلاس NP- کامل [4]

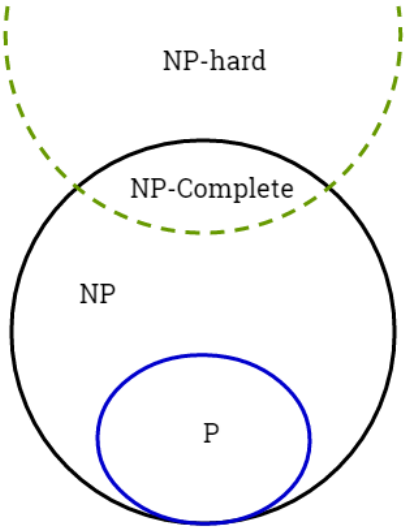
NP- کامل‌ها مسائلی هستند که به سرعت قابل حل نیستند. در نظریه پیچیدگی، مسائل NP-کامل دشوارترین مسائل NP هستند که قبل حل با الگوریتم‌های قطعی با زمان چندجمله‌ای نیستند. برای مشخص کردن این‌که آیا مسأله‌ای در این کلاس قرار دارد یا خیر روش اثباتی وجود ندارد و تنها با متناظر کردن مسأله‌ی جدید با یکی از مسأله‌هایی که قبلا در این کلاس قرار گرفته است به گونه‌ای که راه حل یکی براي دیگری نیز کاربرد داشته باشد، این مسأله جدید نیز در کلاس NP-کامل قرار خواهد گرفت.

تعریف ۲-۱-۶. کلاس NP- سخت [5]

برای حل مسائل NP- سخت، الگوریتم‌های قطعی چندجمله‌ای وجود ندارد. به‌عبارت دیگر، زمان حل آن‌ها با افزایش ابعاد مسأله به‌صورت نمایی افزایش می‌یابد. الگوریتم‌های فراابتکاری به‌عنوان الگوریتم‌های حل مسائل NP- سخت ارائه

[1] decision problems [2] Polynomial [3] Non-deterministic Polynomial [4] NP-complete [5] NP-hard

شده‌اند. این الگوریتم‌ها می‌توانند برای حل این‌گونه مسائل، یک جواب مناسب را در زمانی قابل قبول به‌دست آورند. در شکل ۲-۱ رابطه بین این کلاس‌ها به نمایش گذاشته شده‌است.



شکل ۲-۱: رابطه بین مسائل کلاس P ، NP ، NP-Complete و NP-hard

۲-۱-۳ روش‌های حل مسائل بهینه‌سازی

همان‌طور که بیان شد، مسائل بهینه‌سازی به دو دسته‌ی مسائل بهینه‌سازی ترکیبی و پیوسته تقسیم‌بندی می‌شوند. روش‌های بهینه‌سازی خطی و غیرخطی از جمله روش‌های حل مسائل بهینه‌سازی پیوسته به حساب می‌آیند. روش‌های حل مسائل بهینه‌سازی ترکیبی را نیز می‌توان به دو قسمت الگوریتم‌های دقیق [۱] و الگوریتم‌های ابتکاری [۲] (یا تقریبی) تقسیم‌بندی کرد.

تعریف ۲-۱-۷. االگوریتم‌های دقیق، روش‌هایی هستند که رسیدن به جواب بهینه را تضمین می‌کنند و معمولاً برای حل مسائل با اندازه‌های کوچک یا متوسط، می‌توانند به‌کار گرفته شوند (نظیر روش شاخه و حد [۳]).

تعریف ۲-۱-۸. اگر S کل فضای جواب باشد، آنگاه جواب $s \in S$، یک بهینه محلی است، اگر بهترین کیفیت را نسبت به تمام همسایه‌های خود داشته باشد.

تعریف ۲-۱-۹. الگوریتم‌های ابتکاری، روش‌هایی هستند که رسیدن به جواب بهینه را تضمین نمی‌کنند و تلاشی برای خروج از بهینه محلی نیز انجام نمی‌دهند و برای حل مسائل بهینه‌سازی بزرگ استفاده می‌شوند.

بیشتر مسائل بهینه‌سازی ترکیبی، در کلاس NP- سخت قرار می‌گیرند. بنابراین، برای حل آن‌ها باید از الگوریتم‌های ابتکاری که جواب‌های نزدیک به بهینه را در زمانی قابل قبول به‌دست می‌آورند، استفاده کرد. روش‌های ابتکاری مختلفی

[۱] exact algorithms [۲] heuristic algorithms [۳] Branch & Bound method

برای حل مسائل بهینه‌سازی ترکیبی ابداع شده‌است که در آن‌ها جواب نزدیک به بهینه تولید می‌شود. مشکل این روش‌ها، این است که اکثر آن‌ها فقط در مورد یک مسأله به‌خصوص قابل کاربرد هستند.

تعریف ۲-۱-۱۰. اگر برای حل مسأله‌ای هیچ نوع الگوریتمی با زمان محاسباتی چندجمله‌ای وجود نداشته باشد، مسأله سخت نامیده می‌شود.

بهینه‌سازی غیرخطی پیوسته و روش‌های تقریبی مسائل گسسته، جزء مسائل سخت محسوب می‌شوند.

۲-۱-۴ الگوریتم‌های فراابتکاری

یکی از عیب‌های الگوریتم‌های ابتکاری، تولید یک جواب یا تعداد محدودی از جواب‌ها و توقف آن‌ها در بهینه‌ی محلی با کیفیت پایین است. الگوریتم‌های فراابتکاری، برای حل این مشکلات و عیب‌های روش‌های ابتکاری پیشنهاد شده‌اند.

تعریف ۲-۱-۱۱. الگوریتم‌های فراابتکاری، روش‌هایی هستند که الگوریتم‌های ابتکاری را هدایت و تنظیم می‌کنند تا بتوانند از بهینه‌های محلی عبور کرده و یا حتی‌الامکان در بهینه‌های محلی قرار نگیرند و جوابی نزدیک به بهینه را در زمانی معقول به‌دست آورند.

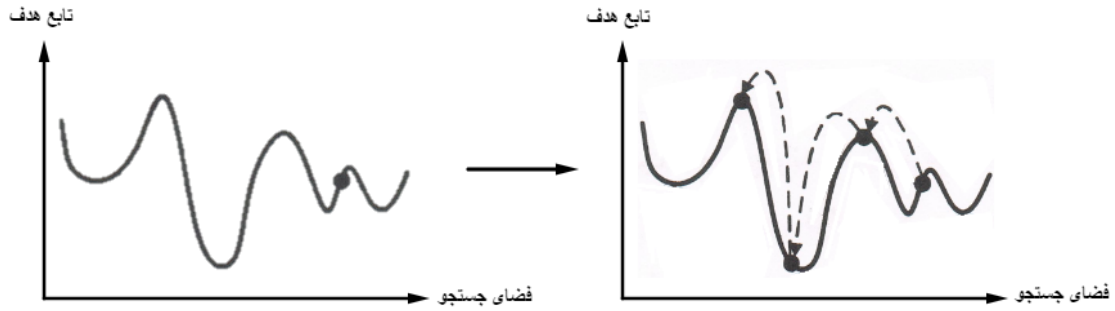
استفاده از الگوریتم‌های فراابتکاری، به‌طور قابل ملاحظه‌ای توانایی یافتن جواب‌هایی با کیفیت بالا را برای مسائل بهینه‌سازی ترکیبی افزایش می‌دهد. به‌عبارت دیگر، یک الگوریتم فراابتکاری، یک روش ابتکاری است که قادر به جستجوی فضای جواب، برای یافتن جواب‌های با کیفیت بالاست. هدف مشترک تمامی الگوریتم‌های فراابتکاری، حل مسائلی است که به مسائل بهینه‌سازی سخت مشهورند.

روش‌های فراابتکاری دارای ویژگی‌های مشترک زیر هستند:

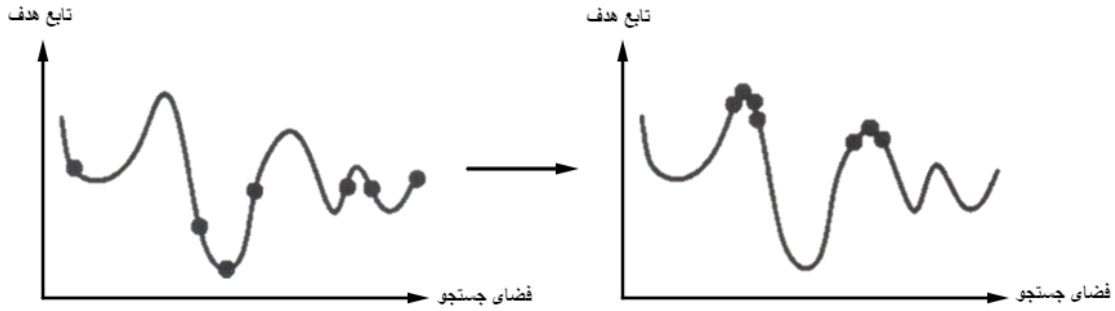
- این روش‌ها همگی تا حدودی احتمالی هستند. چنین رویکردی، این امکان را می‌دهد که از قرار گرفتن الگوریتم در دام بهینه محلی جلوگیری شود.
- این روش‌ها معمولاً برای حل مسائل گسسته ارائه شده‌اند، اما امکان کاربرد در مسائل پیوسته را نیز دارند.
- این روش‌ها غالباً از مفاهیم زیست‌شناسی، رفتارشناسی جانوری و فیزیک الهام گرفته شده‌اند.
- یکی از عیب‌های مشترک در این روش‌ها، دشوار بودن تنظیم و تطبیق پارامترهاست.

از معیارهای مختلفی برای طبقه‌بندی الگوریتم‌های فراابتکاری استفاده می‌شود:

- **مبتنی بر یک جواب و مبتنی بر جمعیت:** الگوریتم‌های مبتنی بر یک جواب، در حین فرآیند جستجو یک جواب را تغییر می‌دهند و این توانایی را دارند که بر روی مناطق محلی جستجو تمرکز کنند (شکل ۲-۲).



شکل ۲-۲: الگوریتم‌های مبتنی بر یک جواب



شکل ۲-۳: الگوریتم‌های مبتنی بر جمعیت

در الگوریتم‌های مبتنی بر جمعیت، در حین جستجو، یک جمعیت از جواب‌ها در نظر گرفته می‌شوند و می‌توانند جستجو را به مناطق مختلفی از فضای جواب هدایت کنند (شکل ۲-۳).

- **الهام گرفته از طبیعت و بدون الهام از طبیعت**: بسیاری از الگوریتم‌های فراابتکاری از طبیعت الهام گرفته‌اند. نظیر الگوریتم‌های تکاملی که از زیست‌شناسی الهام گرفته‌اند، بهینه‌سازی کلونی مورچه‌ها (ACO)[۱] ، کلونی زنبورها (BC)[۲] و بهینه‌سازی گروهی ذرات (PSO)[۳] که از هوش جمعی در گونه‌های مختلف موجودات الهام گرفته‌اند و تبرید شبیه‌سازی شده (SA)[۴] که از فیزیک الهام گرفته است.

- **با حافظه و بدون حافظه**: برخی از الگوریتم‌های فراابتکاری، فاقد حافظه هستند. به این معنا که از اطلاعات به‌دست آمده در حین جستجو استفاده نمی‌کنند. به‌عنوان نمونه‌ای از این الگوریتم‌ها می‌توان به تبرید شبیه‌سازی شده اشاره کرد. این در حالی است که برخی از الگوریتم‌های فراابتکاری نظیر جستجوی ممنوعه (TS)[۵] ، از حافظه استفاده می‌کنند. این حافظه، اطلاعات به‌دست آمده در طول جستجو را در خود ذخیره می‌کند.

- **قطعی و احتمالی**: یک الگوریتم فراابتکاری قطعی نظیر جستجوی ممنوعه، مسأله را با استفاده از تصمیمات قطعی حل می‌کند، اما در الگوریتم‌های فراابتکاری احتمالی نظیر تبرید شبیه‌سازی شده، یک سری قوانین احتمالی در طول جستجو مورد استفاده قرار می‌گیرد. در الگوریتم‌های قطعی، با شروع از یک جواب اولیه یکسان، به یک جواب

---

[۱]Ant Colony Optimization [۲]Bee Colony [۳]Particle Swarm Optimization [۴]Simulated Annealing [۵]Tabu Search

نهایی مشابه می‌رسیم، در حالیکه الگوریتم‌های احتمالی ممکن است از یک جواب اولیه مشابه به جواب‌های نهایی مختلفی برسند. این ویژگی‌ها باید در زمان تحلیل عملکرد الگوریتم‌های فراابتکاری مورد توجه قرا گیرند.

- **تکرارشونده و حریصانه**: الگوریتم‌های تکرارشونده با یک جواب (یا جمعیتی از جواب‌های) کامل شروع می‌کنند و آن را در هر تکرار با استفاده از برخی عملگرهای جستجو تغییر می‌دهند. الگوریتم‌های حریصانه با یک جواب خالی شروع و در هر گام، یک متغیر تصمیم از مسأله را مقداردهی می‌کنند تا این‌که یک جواب کامل به‌دست آید. بیشتر الگوریتم‌های فراابتکاری تکرارشونده هستند.

۲-۱-۴-۱ الگوریتم‌های فراابتکاری مبتنی بر یک جواب

الگوریتم‌های مبتنی بر یک جواب، به‌طور تکرار شونده از جای‌گیزینی و تولید یک جواب جاری استفاده می‌کنند. در مرحله‌ی تولید،یک مجموعه از جواب‌های کاندید از جواب فعلی s تولید می‌شوند. این مجموعه که آن را $C(s)$ می‌نامیم، معمولاً به‌وسیله انتقال‌های محلی به‌دست می‌آیند. در مرحله‌ی جای‌گزینی، انتخاب از جواب‌های کاندید صورت می‌گیرد، تا جواب فعلی به وسیله‌ی یک جواب جدید جای‌گزین شود. به این معنا که جواب $s' \in C(s)$، به‌عنوان جواب جدید انتخاب می‌شود. این فرآیند تا زمان رسیدن به شرط خاتمه ی از پیش تعیین شده تکرار می شود.

الگوریتم ۲-۱ شبه کد الگوریتم‌های فراابتکاری مبتنی بر یک جواب

Require: initial solution s_0.
Ensure: best solution found.
$t = 0$
while Stoping criteria satisfied **do**
  Generate $(C(s_t))$
  s_{t+1} = Select $(C(s_t))$
  $t = t + 1$
end while

تعریف همسایگی، معمولاً در طراحی تمامی الگوریتم‌های مبتنی بر یک جواب وجود دارد. ساختار همسایگی، نقش اساسی در کیفیت عملکرد این الگوریتم‌ها بازی می‌کند. اگر ساختار همسایگی برای مسأله مناسب نباشد، با هیچ یک از الگوریتم‌های مبتنی بر یک جواب نمی‌توان حل کرد.

مجموعه جواب‌های همسایه s از کل فضای جواب S، به‌صورت $N : S \to ۲^s$ تعریف می‌شود که در آن $N(s) \subset S$. یک جواب s' در همسایگی s ($s' \in N(s)$)، همسایه‌ی s نام دارد. یک جواب همسایهِ‌ی s' با بکارگیری عملگر حرکت [۱] ، جایگزین جواب فعلی s می‌شود. عملگر حرکت، یک به هم ریختگی کوچک در s ایجاد می‌کند.

تعریف ۲-۱-۱۲. هر عملیاتی که برای تعریف یک همسایگی از هر جواب مفروض استفاده شود، را یک حرکت می‌نامند.

[۱] move

خصوصیت اصلی یک همسایه، محلی بودن [1] آن است. محلی بودن، اثری است بر روی جواب، که هنگام اجرای حرکت (به هم ریختگی) ایجاد می‌شود.

- یک جواب دارای محلی بودن قوی است اگر با انجام تغییر کوچکی بر روی جواب، برازندگی آن نیز به میزان کمی تغییر کند.
- محلی بودن ضعیف، مربوط به ایجاد آثار بزرگ بر روی برازندگی جواب، در اثر تغییرهای کم بر روی جواب است که در این حالت جستجو تبدیل به یک جستجوی تقریباً تصادفی می‌شود.

منشأ بسیاری از الگوریتم‌های مبتنی بر یک جواب، جستجوی محلی (LS) [2] است. جستجوی محلی یک روش ابتکاری است که از یک جواب اولیه شدنی شروع می‌کند و در هر تکرار، جواب فعلی را با یک همسایه جای‌گزین می‌کند. در این روش، جواب همسایه‌ای که جای‌گزین جواب فعلی می‌شود، باید تابع هدف را بهبود بخشد. جستجو زمانی پایان می‌یابد که همسایه‌های کاندید، بدتر از جواب فعلی باشند و این بدان معناست که بهینه محلی یافت شده‌است. یکی از محدودیت‌های جستجوی محلی این است که معمولاً در یک جواب متوسط و نه خوب متوقف می‌شود. در LS پیچیدگی محاسباتی به توانایی و قدرت حرکت‌های انجام شده در هر تکرار بستگی دارد.

الگوریتم ۲-۲ شبه کد جستجوی محلی

Require: initial solution s_0.
Ensure: local optima found.

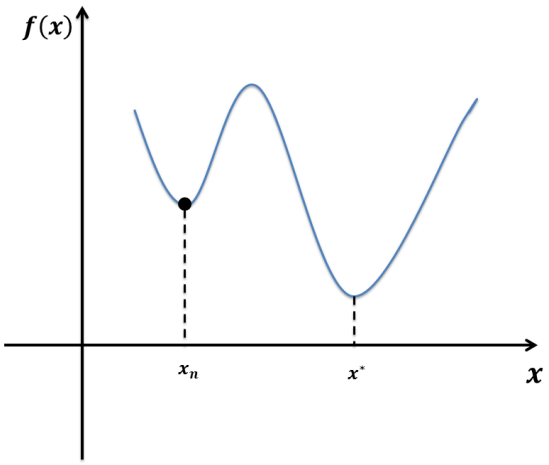
```
s = s_0
while not temination criterion do
  Generate (N(s))
  if there is no better neighbor then
    stop
  else
    s = s'
  end if
end while
```

مثال ۲-۱-۱۳. مسأله حداقل‌سازی تابع $f(x)$روی یک مجموعه محدود از نقاط را در نظر بگیرید. مشکل گیر کردن در بهینه محلی، زمانی اتفاق می‌افتد که الگوریتم به نقطه بهینه محلی (x_n) برسد. در این نقطه الگوریتم به جستجوی همسایگی پرداخته و یک همسایه را انتخاب می‌نماید.

زمانی که الگوریتم در بهینه محلی قرار دارد، نقاط همسایه آن دارای مقدار تابع بدتر از خود نقطه بهینه محلی می‌باشند. از روی اجبار، الگوریتم به بهترین نقطه بین آن‌ها حرکت می‌کند. بعد از حرکت به آن نقطه دوباره به جستجوی همسایگی می‌پردازد و با توجه به اینکه بهینه محلی قبلی در همسایگی نقطه جدید قرار دارد مجدداً به بهینه محلی برمی‌گردد. در این

[1] locality [2] Local Search

حالت الگوریتم در یک دور افتاده است و از آن خارج نمی‌شود، این وضعیت به گیر کردن در بهینه محلی معروف است. شکل ۲-۴ اسیر شدن الگوریتم در بهینه محلی را نشان می‌دهد [۲۲].



شکل ۲-۴: به دام افتادن الگوریتم در جواب بهینه محلی (x_n)

طراحی و پیاده‌سازی جستجوی محلی آسان بوده و جواب‌های خوبی را در زمان کوتاه به‌دست می‌دهد. به همین دلیل، این روش در عمل بسیار استفاده می‌شود. یکی از معایب مهم LS ، همگرایی به بهینه محلی است. علاوه بر این، جستجوی محلی نسبت به جواب اولیه بسیار حساس است؛ به این معنا که با شروع از جواب‌های اولیه‌ی مختلف، معمولاً جواب‌های نهایی متفاوتی به‌دست می‌آید. اگر تعداد بهینه‌های محلی در فضای جستجو کم باشد، جستجوی محلی خوب کار می‌کند. الگوریتم‌های فراابتکاری، برای جلوگیری از همگرایی LS به بهینه محلی ارائه شده‌اند. استراتژی‌های به‌کار رفته در این الگوریتم‌ها برای جلوگیری از همگرایی در بهینه‌ی محلی را می‌توان در چهار دسته به‌صورت زیر تقسیم‌بندی کرد:

- شروع از جواب‌های اولیه‌ی متفاوت به‌صورت تکرارشونده؛
- پذیرش جواب‌های بدون بهبود؛
- تغییر همسایگی؛
- تغییر تابع هدف یا داده‌های ورودی مسأله.

۲-۱-۴-۲ الگوریتم‌های فراابتکاری مبتنی بر جمعیت

بیشتر الگوریتم‌های مبتنی بر جمعیت، نظیر الگوریتم ژنتیک، کلونی مورچگان، کلونی زنبورها و ... از طبیعت الهام گرفته شده‌اند. الگوریتم‌های مبتنی بر جمعیت را می‌توان به‌صورت یک فرآیند بهبود تکرارشونده در یک جمعیت از جواب‌ها تعریف کرد. این الگوریتم‌ها، با یک جمعیت اولیه از جواب‌ها شروع به‌کار می‌کنند. در مرحله‌ی تولید، یک جمعیت جدید

از جواب‌ها ایجاد می‌شود. در مرحله‌ی جای‌گزینی، این جمعیت از جواب‌ها به‌وسیله‌ی روش‌های مختلف انتخاب، با جمعیت فعلی ادغام می‌شود. این فرآیند تا جایی تکرار می‌شود که شرط خاتمه مشاهده شود.

الگوریتم ۲-۳ شبه کد الگوریتم‌های فراابتکاری مبتنی بر جمعیت

Require: initial population P_0.
Ensure: best solution(s) found.
$P = P_0$
$t = 0$
while Stoping criteria satisfied **do**
Generate(P'_t)
P_{t+1} = Select-Population($P_t \cup P'_t$)
$t = t + 1$
end while

یکی از انواع الگوریتم‌های مبتنی بر جمعیت، الگوریتم‌های تکاملی [۱] هستند. تکامل به معنی پیشرفت تدریجی است. امروزه یکی از مهمترین زمینه‌های تحقیق و پژوهش، توسعه‌ی روش‌های جستجو براساس اصول تکامل طبیعی می‌باشد.تکامل در طبیعت را می‌توان به‌صورت زیر تشریح کرد. جمعیتی از افراد رادر یک محیط زیست در نظر بگیرید که هر یک از این افراد برای بقا و تولیدمثل تلاش می‌کنند. هر فرد در این جمعیت دارای یک برازندگی [۲] است که برتری وی را نسبت به سایرین نشان می‌دهد. این برازندگی توسط طبیعت تعیین می‌شود. در فرآیند حل مسأله، مجموعه‌ای از جواب‌های کاندید در اختیار داریم که کیفیت این جواب‌ها شانس آن‌ها را در باقی ماندن و استفاده شدن در ساخت جواب‌های کاندید دیگر تعیین می‌کند.

ایده‌ی به‌کار بردن اصول تکامل در طبیعت برای حل مسأله، به دهه‌ی چهل میلادی، یعنی بسیار قبل از پیشرفت کامپیوترها باز می‌گردد. در اوایل سال ۱۹۴۸، تورینگ [۳]، جستجوی ژنتیکی یا تکاملی را پیشنهاد کرد. در طول دهه‌ی شصت سه نوع پیاده‌سازی مختلف از ایده‌ی اولیه توسعه داده شد. در آمریکا، فوگل [۴] و همکارانش برنامه‌ریزی تکاملی (EP) [۵] را معرفی کردند. در همان زمان، هالند [۶]، این روش را الگوریتم ژنتیک نامید و در آلمان، ریچنبرگ [۷]، استراتژی‌های تکاملی (ES) [۸] را ابداع کرد. این زمینه‌ها در طول پانزده سال به‌صورت جداگانه توسعه داده شدند. در اوایل دهه‌ی نود، چهارمین جریان به نام برنامه‌ریزی ژنتیک (GP) [۹] توسط کوزا [۱۰] ارائه شد. امروزه تمامی این زمینه‌ها با عنوان محاسبات تکاملی [۱۱] شناخته می‌شوند و به الگوریتم‌های تکاملی مشهورند [۱]. الگوریتم‌های تکاملی در ابتدا به دلیل هزینه‌ی بالای اجرایشان با اقبال کمی مواجه شدند، اما در دو دهه‌ی اخیر به موجب افزایش قدرت محاسبات رایانه‌ها گسترش قابل توجه‌ای یافته‌اند. معروف‌ترین الگوریتم تکاملی، الگوریتم ژنتیک است که در بخش ۲-۳ به تفصیل آن‌را مورد بررسی قرار می‌دهیم.

[۱] evolutionary algorithms [۲] fitness [۳] turing [۴] fogel [۵] Evolutionary Programming
[۶] holland [۷] rechenberg [۸] Evolutionary Strategies [۹] Genetic Programming [۱۰] koza
[۱۱] Evolutionary Computing

۲-۲ الگوریتم جستجوی ممنوعه (TS)

در این بخش، ابتدا نگاهی اجمالی به الگوریتم جستجوی ممنوعه و خصوصیات آن می‌اندازیم و سپس کاربرد این الگوریتم در حل مسأله مکانیابی هاب تک تخصیصی بدون ظرفیت را مورد بررسی قرار می‌دهیم.

۲-۲-۱ مقدمه و تاریخچه

گلاور [۱] که یکی از سرشناسان حوزه هوش مصنوعی (AI)[۲] به‌شمار می‌رود، به‌طور منسجم به مطالعه فرایند حل مسائل بهینه‌سازی در مغز انسان ازحدود سال ۱۹۶۷ تا سال ۱۹۸۶ پرداخته است و برمبنای یافته هایش جستجوی ممنوعه (TS) [۳] را برای اولین بار در سال ۱۹۸۶ [۲۳] معرفی کرد. واژه‌ی متاهیوریستیک به معنای فراابتکاری اولین بار در این مقاله به‌کار برده شده‌است. از آنجا که ساختار مغز انسان بسیار پیچیده است، شاید به جرأت بتوان گفت که الگوریتم جستجوی ممنوعه نیز پیچیده‌ترین و هوشمندترین الگوریتم فراابتکاری است که تاکنون متولد شده‌است. بنابراین، اگر کسی که قصد استفاده از این الگوریتم را دارد، بدون اشراف کامل بر مبانی آن اقدام به کدنویسی کند معمولاً نتایج بدی به‌دست می‌آورد. از جمله مواردی که در این الگوریتم، متناظر با ساختار مغز انسان وجود دارد، می‌توان به یادگیری، فراموشی، حافظه کوتاه مدت و حافظه بلند مدت، استنتاج و... اشاره کرد [۲۴]. در سال ۱۹۹۷، اولین کتابی که کاملاً به جستجوی ممنوعه اختصاص داشت توسط گلاور و لاگونا منتشر شد [۲۵].

واژه‌ی تابو [۴] از تُنگان [۵] زبان مردم جزایر پلینزی در اقیانوس آرام گرفته شده‌است. این واژه به معنای شیء مقدسی است که به‌دلیل قداست نباید آن را لمس کرد. براساس واژه‌نامه‌ی وبستر [۶] ، امروزه این واژه به معنای "ممنوعیت چیزی که دارای ریسک است" به‌کار می‌رود که با تکنیک جستجوی ممنوعه کاملاً سازگار است. ریسکی که در الگوریتم جستجوی ممنوعه از آن اجتناب می‌شود، خطر مسیرهای نامناسب است.

در سال‌های اخیر، مجله‌های مختلفی در طیف گسترده‌ای از زمینه‌ها، مقاله‌ها و مطالعه‌هایی را چاپ کرده‌اند که موفقیت TS را در بسیاری از مسائل نشان می‌دهد. جستجوی ممنوعه، به‌طور قابل ملاحظه‌ای توانایی ما را در حل مسائل عملی افزایش می‌دهد. جدول ۲-۱ برخی از کاربردهای TS را ارائه می‌کند.

۲-۲-۲ ساختارهای حافظه و تابو

جستجوی ممنوعه برای این‌که به‌عنوان یک روش هوشمندانه حل مسأله شناخته شود باید از حافظه سازگار [۷] و جستجوی واکنش دهنده [۸] استفاده کند.

---

[۱] glover [۲] Artificial Intelligence [۳] Tabu Searchs [۴] Tabu (Taboo) [۵] tongan [۶] webster
[۷] adaptive memory [۸] responsive exploration

جدول ۲-۱: کاربردهای جستجوی ممنوعه [۱]

نوع مسأله	نام مسأله
مسیریابی	فروشنده‌ی دوره‌گرد، مسیریابی وسیله نقلیه، مسیریابی پنجره‌ی زمانی
مکان‌یابی و تخصیص	جایابی تسهیلات، تخصیص درجه ۲
ساخت و تولید	تولید سلولی، تولید انعطاف‌پذیر، زمان‌بندی کارگاهی
زمان‌بندی	زمان‌بندی پروژه، تولید جدول‌های زمانی
داده‌کاوی	خوشه‌بندی
طراحی	طراحی شبکه ظرفیت‌دار

- حافظه سازگار در TS ، جستجوی فضای جواب را اثربخش و اقتصادی می‌کند به‌طوریکه جواب‌های محلی در جستجوی ممنوعه به‌وسیله‌ی اطلاعات جمع‌آوری شده در حین جستجو، هدایت می‌شوند.
- جستجوی واکنش‌دهنده مبنای اساسی جستجوی هوشمند است، یعنی استفاده از ویژگی‌های جواب خوب برای جستجوی نواحی امیدبخش جدید. تأکید بر استفاده از جستجوی واکنش‌دهنده در جستجوی ممنوعه، به این دلیل است که یک انتخاب استراتژیک بد(یعنی یک انتخاب براساس مکانیزم از پیش تعیین شده) می‌تواند از یک انتخاب خوب تصادفی، اطلاعات بیشتری را فراهم کند.

حافظه، در طول جستجو به ایجاد تفاوت میان کیفیت جواب‌های بازدید شده کمک می‌کند و حرکت‌هایی را که منجر به جواب‌های خوب می‌شوند تشویق می‌کند. حافظه‌ای که در جستجوی ممنوعه مورد استفاده قرار می‌گیرد، به دو نوع حافظه جامع و حافظه وصفی تقسیم‌بندی می‌شود.

- **حافظه جامع**: حافظه‌ای است که همه‌ی جواب‌هایی که در طی جستجو یافت شده‌اند، از جمله جواب‌های خیلی خوب را ذخیره می‌کند. حالت گسترش یافته‌ای از این حافظه، همسایه‌های بسیار جذاب از این جواب‌های خوب که تاکنون کشف نشده‌اند را نیز ذخیره می‌کند. حفظ کردن جواب‌های خوب یا همسایه‌های جذاب آن‌ها برای بهبود جستجوی محلی مورد استفاده قرار می‌گیرند.
- **حافظه وصفی**: جستجوی محلی از این حافظه برای تعیین جهت حرکت در فضای حل استفاده می‌کند. این نوع حافظه، اطلاعات مربوط به ویژگی‌های متغیر جواب‌های هر تکرار را ثبت می‌کند. به‌عنوان مثال در مسائل شبکه، ویژگی‌های مورد بررسی می‌تواند شامل گره‌ها یا یال‌هایی باشد که با یک مکانیزم به حافظه اضافه، حذف یا تعویض می‌شوند.

حافظه‌های جامع و وصفی مکمل یکدیگرند. بدین صورت که حافظه جامع، تعداد همسایگی‌ها را در طول جستجوی محلی با در نظر گرفتن جواب‌های خیلی خوب افزایش می‌دهد و حافظه وصفی، با غربال کردن یا ممنوع کردن بعضی از حرکات، آن را کاهش می‌دهد. دو عنصر اصلی جستجوی ممنوعه، فضای جستجو و ساختار همسایگی است.

تعریف ۲-۲-۱. فضای جستجو: فضایی از جواب‌های شدنی است که می‌توان آن‌ها را حین جستجو بازدید کرد و در اینجا با S نمایش داده می‌شود.

محدود کردن فضای جستجو به جواب‌های شدنی همیشه ایده‌ی خوبی نیست. اجازه دادن به جستجو برای حرکت به جواب‌های نشدنی مطلوب در برخی موارد لازم و ضروری است.

تعریف ۲-۲-۲. همسایگی‌های جواب فعلی $s \in S$ ، زیرمجموعه‌ای از فضای جستجو است که به‌صورت زیر تعریف می‌شود:

$$N(s) = \{\text{جواب‌های بدست آمده با به‌کار گیری یک انتقال از s}\}$$

مثال ۲-۲-۳. اگر S مجموعه‌ای از بردارهای صفر و یک و $s \in S$ باشد، یک مجموعه همسایگی $N(s)$ برای s، مجموعه جواب‌های $s \in S$ است که با تغییر وضعیت تنها یکی از مؤلفه‌ها از صفر به یک یا بالعکس بدست آمده‌اند. مثلا اگر $s = (۰, ۰, ۱, ۰)$ باشد آنگاه:

$$N(s) = \{(۱, ۰, ۱, ۰)(۰, ۱, ۱, ۰)(۰, ۰, ۰, ۰)(۰, ۰, ۱, ۱)\}$$

یکی از نکات مهم در مورد TS ، استفاده از حافظه کوتاه‌مدت و بلندمدت است. اثر هر دو نوع حافظه را می‌توان در تغییر همسایگی $N(s)$ برای جواب کنونی s مشاهده کرد.

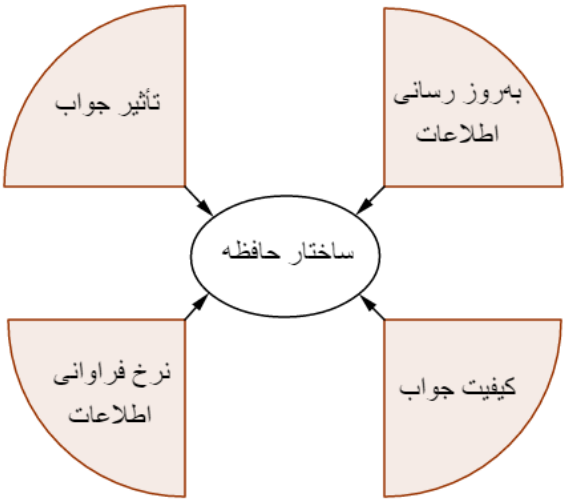
تعریف ۲-۲-۴. همسایگی اصلاح شده: نتیجه‌ی نگه‌داری سوابق مشاهده شده در حین جستجو است که با $N^*(s)$ نمایش می‌دهیم وبه دلیل استفاده از حافظه، مسیر حرکت از یک جواب به جواب بعدی را تعیین می‌کند.

در استراتژی‌های مبتنی برحافظه‌ی کوتاه‌مدت، $N^*(s)$ به‌طور مشخص زیرمجموعه‌ای از $N(s)$ است در واقع $N^*(s)$ از $N(s)$ با توجه به لیست ممنوع بدست می‌آید. در فرآیند مبتنی بر این حافظه، ممکن است به یک جواب s اجازه داده شود که بیش از یک بار بازدید شود، اما این احتمال وجود دارد که همسایگی $N^*(s)$ در هر زمان متفاوت باشد. هسته‌ی اصلی جستجوی ممنوعه در فرآیند حافظه کوتاه مدت مجسم می‌شود.

در استراتژی‌های مبتنی بر حافظه بلندمدت، $N^*(s)$ ممکن است شامل جواب‌هایی به غیر از جواب‌های معمول در $N(s)$ نیز باشد. با اضافه شدن این حافظه به فرآیند، احتمال این‌که زیرمجموعه‌ای از S ، به‌طور تکراری چندین بار انتخاب شوند تقریباً وجود نخواهد داشت.

ساختار حافظه‌های مورد استفاده در روش جستجوی ممنوعه براساس چهار عنصر شکل می‌گیرد. این عناصر در شکل ۲-۵ نشان داده شده‌اند.

۱. **عنصر کیفیت جواب:** از این‌گونه حافظه‌ها به‌منظور شناسایی وجوه مشترک جواب‌های مناسب و یا مسیرهایی که به چنین جواب‌هایی منجر می‌شوند، استفاده می‌شود. در واقع، این عنصر به این موضوع اشاره دارد که در الگوریتم، جواب‌هایی که طی فرآیند جستجو یافت می‌شوند، می‌توانند ارزش‌هایی یکسانی نداشته باشند.



شکل ۲-۵: ساختار حافظه در الگوریتم TS

۲. **عنصر تأثیر جواب:** به بررسی اثر هر یک از حالت‌های انتخاب جهت جستجو که در زمان حل صورت می‌گیرد روی کیفیت و ساختار حل مسأله می‌پردازد.

۳. **عنصر به‌روز سانی اطلاعات:** حافظه‌ای که بر اساس به‌روز رسانی کار می‌کند، یا بعبارتی حافظه‌ی مبتنی بر تأخر [۱] ، نوعی حافظه کوتاه‌مدت است و همان‌طور که از نام آن مشخص است ویژگی‌های جواب‌هایی را که اخیراً تغییر کرده‌اند نگه‌داری می‌کند. به بیان دیگر، تعداد تکرارهای متوالی که اجزای مختلف جواب در جواب فعلی حضور می‌یابند را ذخیره می‌کند.

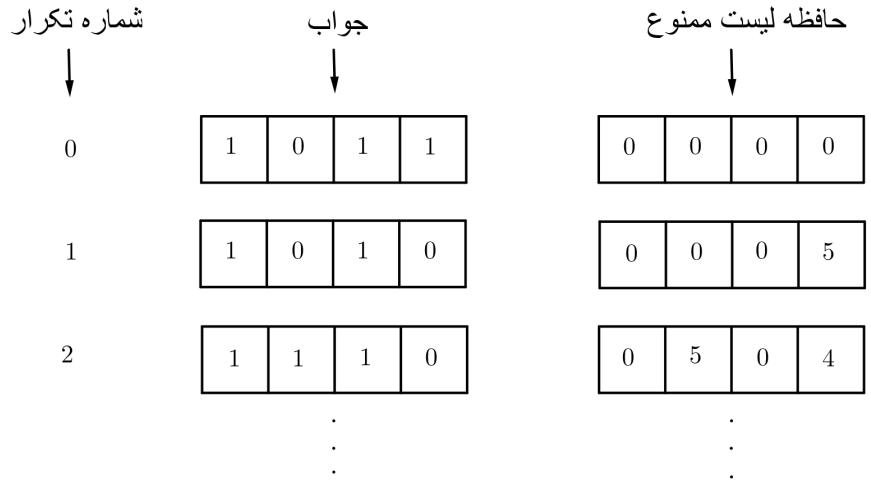
۴. **عنصر نرخ فراوانی اطلاعات:** حافظه‌ای که بر اساس نرخ فراوانی کار می‌کند، یا بعبارتی حافظه‌ی مبتنی بر تناوب [۲] ، نوعی حافظه بلندمدت است که تعداد کل تکرارهایی را که اجزای جواب در جواب فعلی یا در حرکت‌های انتخابی وجود داشته‌اند را ذخیره می‌کند.

در بخش ۲-۱-۴-۱ توضیحاتی در مورد الگوریتم جستجوی محلی (LS) ارائه شد. جستجوی ممنوعه یک توسعه روی روش‌های LS کلاسیک به‌حساب می‌آید. یکی از عناصر متمایز کننده‌ی TS از LS ، ساختارهای تابو و حافظه‌ی آن است. در حقیقت، TS را می‌توان ترکیب LS با حافظه‌ی کوتاه مدت دانست. ساختارهای تابو، برای جلوگیری از ایجاد چرخه و فرار از بهینه‌ی محلی مورد استفاده قرار می‌گیرند. در نتیجه، ابزاری موردنیاز است که از بازگشت جستجو به مکان قبلی خود جلوگیری کند. این ابزار توسط تعریف حرکت‌های ممنوعه (غیرمجاز) به‌دست می‌آید.

تعریف ۲-۲-۵. حرکت‌های ممنوعه، حرکت‌هایی هستند که اثر حرکت‌های اخیر را خنثی می‌کنند.

رویکردی که گلاور با نام جستجوی محلی در سال ۱۹۸۶ ارائه کرد، در واقع به جستجوی محلی اجازه می‌دهد تا بر بهینه‌ی محلی غلبه کند. اصول و مبانی TS ، این فرصت را برای جستجوی محلی فراهم می‌کند که در هرجایی که به

[۱] recency-based memory [۲] frequency-based memory



شکل ۲-۶: نحوه‌ی شکل‌گیری لیست ممنوع

بهینه محلی رسید با استفاده از حافظه‌ها که لیست ممنوعه [۱] نام دارند از بهینه‌ی محلی خارج و همچنین از برگشت به آن‌ها جلوگیری شود.

تعریف ۲-۲-۶. لیست ممنوعه، فهرستی از حرکت‌های ممنوعه است که برای حافظه‌های مورد استفاده در الگوریتم جستجوی ممنوعه تعریف می‌شود.

به بیان دیگر، این لیست شامل تعدادی از حرکت‌های اخیر جستجوگر است که اجازه رجوع به آن‌ها در حرکت فعلی وجود ندارد . پس از انجام هر حرکت، جواب فعلی در بالای لیست ممنوعه قرار گرفته و چنانچه لیست پر باشد، قدیمی‌ترین جواب لیست از آن خارج می‌گردد. طول لیست ممنوعه می‌تواند ثابت یا متغیر باشد.

تعریف ۲-۲-۷. هر حرکت ممنوعه که در لیست ممنوعه وارد شود برای تعداد تکرار معینی، ممنوع خواهد بود و پس از آن مجدداً قابل بهره‌برداری است. این تعداد تکرار، مدت ممنوعه [۲] نام دارد.

در مثال ۲-۲-۸ نحوه‌ی شکل‌گیری لیست ممنوعه را بیان می‌کنیم.

مثال ۲-۲-۸. فرض کنید جواب مسأله‌ای فقط شامل صفر و یک است و لیست ممنوعه نیز در ابتدا صفر باشد، مدت ممنوع بودن هر تغییر نیز تا ۵ تکرار است. حرکت جواب جاری به جواب همسایه با تغییر یکی از عنصرهای جواب جاری از صفر به یک یا برعکس تعریف می‌شود. بعد از تغییر، عنصر متناظر با آن مؤلفه در حافظه‌ی لیست ممنوعه، عدد ۵ است و در هر تکرار یک واحد از آن کاهش می‌یابد. همان‌طور که در شکل ۲-۶ می‌بینید عنصر چهارم جواب جاری در تکرار اول از یک به صفر تغییر یافته است، بنابراین عنصر چهارم از حافظه لیست ممنوع عدد ۵ است و در تکرار دوم یک واحد کاهش پیدا می‌کند.

---
[۱] tabu list [۲] tabu tenure

گاهی اوقات در TS ، ساختارهای ممنوعه بسیار قدرتمند می‌شوند و ممکن است حتی هنگامی‌که هیچ خطر سیکلی وجود ندارد، از حرکت‌های مطلوب جلوگیری کنند و یا منجر به ایستایی و رکود کلی فرآیند جستجو شوند. در این حالت نیاز است که از یک ابزار الگوریتمیک استفاده کنیم تا تابوها را لغو و باطل کند. این ابزار، معیار تنفس [1] نام دارند. قانون کلیدی در معیار تنفس، این است که اگر چرخه اتفاق نیفتد می‌توان از تابوها صرف نظر کرد. ساده‌ترین و رایج‌ترین معیار تنفس مورد استفاده، اجازه دادن به حرکت ممنوعه‌ای است که از بهترین جواب یافت شده تاکنون بهتر است.

۲-۲-۳ معیارهای توقف

رایجترین معیارهای توقفی که در TS مورد استفاده قرار می‌گیرند، عبارتند از:

- پس از تعداد معینی تکرار؛
- میزان ثابتی از زمان CPU ؛
- پس از تعداد تکرارهای معینی که هیچ بهبودی در تابع هدف ایجاد نکنند (معیار توقفی که در اکثر پیاده‌سازی‌ها استفاده شده‌است.)؛
- هنگامی‌که تابع هدف به مقدار آستانه‌ای از پیش تعیین شده نزدیک شود.

در روش‌های پیچیده‌تر جستجوی ممنوعه، جستجو معمولاً پس از اتمام چندین مرحله متوقف می‌شود که هر مرحله براساس یکی از معیارهای بالا متوقف می‌شود.

۲-۲-۴ ساختار کلی جستجوی ممنوعه

برای رسیدن به جواب بهینه در یک مسئله بهینه‌سازی، الگوریتم جستجوی ممنوعه ابتدا از یک جواب اولیه شروع به حرکت می‌کند. سپس الگوریتم بهترین جواب همسایه را از میان همسایه‌های جواب فعلی انتخاب می‌کند. در صورتی که این جواب در فهرست ممنوعه قرار نداشته باشد، الگوریتم به جواب همسایه حرکت می‌کند؛ در غیراین‌صورت الگوریتم معیاری به نام معیار تنفس را چک خواهد کرد. بر اساس معیار تنفس اگر جواب همسایه از بهترین جواب یافت شده تاکنون بهتر باشد، الگوریتم به آن حرکت خواهد کرد، حتی اگر آن جواب در فهرست ممنوعه باشد. پس از حرکت الگوریتم به جواب همسایه، فهرست ممنوعه به‌روز رسانی می‌شود؛ به این معنا که حرکت قبل که بوسیله‌ی آن به جواب همسایه حرکت کردیم در فهرست ممنوعه قرار داده می‌شود تا از بازگشت مجدد الگوریتم به آن جواب و ایجاد سیکل جلوگیری شود. در واقع فهرست ممنوعه ابزاری در الگوریتم جستجوی ممنوعه‌است که توسط آن از قرار گرفتن الگوریتم در بهینه‌ی محلی جلوگیری می‌گردد. پس از قرار دادن حرکت قبلی در فهرست ممنوعه، تعدادی از حرکت‌هایی که قبلاً در فهرست ممنوعه قرار گرفته بودند از فهرست

[1] aspiration criteria

خارج می‌شوند. مدت زمانی که حرکت‌ها در فهرست ممنوعه قرار می‌گیرند توسط یک پارامتر که مدت ممنوعه نام دارد تعیین می‌شود. حرکت از جواب فعلی به جواب همسایه تا جایی ادامه می‌یابد که شرط خاتمه دیده شود.

الگوریتم ۲-۴ شبه کد الگوریتم جستجوی ممنوع

ورودی: S و تابع f

خروجی: مینیمم تابع f

۱: یک جواب اولیه s در S انتخاب کنید و قرار دهید: $s^* = s$ ، $f^* = f(s)$، در این گام لیست ممنوع تهی است $(T = \varnothing)$.

۲: تا زمانیکه شرط خاتمه برقرار شود گام‌های زیر را انجام دهید:

۳: بهترین s' را در مجموعه $(N(s) - T)$ بیابید و قرار دهید $s = s'$.

۴: اگر $f(s') < f(s)$ سپس $s^* = s'$.

۵: لیست ممنوع بهنگام شود.

در ادامه مثالی با روش جستجوی ممنوعه حل خواهد شد که به روشن‌تر شدن مراحل این روش کمک خواهد کرد [۲۲].

مثال ۲-۲-۹. یک مسأله زمان‌بندی[۱] ساده را با یک ماشین و ۴ کار j_1، j_2، j_3، j_4 در نظر بگیرید. هدف این مسأله کمینه کردن مجموع تأخیرها است. داده‌های این مسأله به شرح زیر می‌باشند:

جدول ۲-۲: داده‌های مربوط به مسأله نمونه زمان‌بندی

کارها(j_j)	۱	۲	۳	۴
زمان انجام کار به دقیقه (p_i)	۱۰	۱۰	۱۳	۴
مهلت تحویل کار به دقیقه (d_i)	۴	۲	۱	۱۲
وزن کار (W_i)	۱۴	۱۲	۱	۱۲

هر جواب از این مسأله به شکل یک بردار چهارتایی از ارقام ۱، ۲، ۳، ۴ نشان داده می‌شود. مثلاً اگر به ترتیب کارهای j_3، j_2، j_1 و در انتها نیز j_4 انجام شوند، جواب مربوطه به‌صورت $(۳, ۲, ۱, ۴^{\rightarrow})$ نشان داده می‌شود.

اگر تأخیر نظیر j_j را T_i بنامیم، تابع هدف به شکل زیر محاسبه می‌شود:

$$F = \sum_i W_i T_i$$

برای طراح مسأله مهم است که تابع هدف مقداری کمتر از ۴۱۰ داشته باشد. لذا، از همین موضوع می‌توان به‌عنوان شرط توقف استفاده کرد. طول فهرست ممنوع نیز برابر ۲ داده شده است.

[۱] Scheduling

تکرار ۱: جواب دلخواهی مثل $(۲, ۱, ۴, ۳)$ را به‌عنوان جواب اولیه $x_۱$ انتخاب می‌کنیم:

$$T_۲ = p_۲ - d_۲ = ۱۰ - ۲ = ۸ \Rightarrow W_۲T_۲ = ۹۶$$

$$T_۱ = ۲۰ - ۴ = ۱۶ \quad \Rightarrow W_۱T_۱ = ۲۲۴$$

$$T_۴ = ۲۴ - ۱۲ = ۱۲ \quad \Rightarrow W_۴T_۴ = ۱۴۴$$

$$T_۳ = ۳۷ - ۱ = ۳۶ \quad \Rightarrow W_۳T_۳ = ۳۶$$

$$\Rightarrow \sum_i W_iT_i = ۵۰۰$$

اکنون باید جواب‌های همسایه نظیر این جواب پیدا شوند. در این‌جا جواب‌های همسایه هر جواب را با جابه‌جایی ترتیب هر دو کار مجاور (پشت سر هم) در آن جواب بدست می‌آوریم. بنابراین جواب‌های مجاور $x_۱$ عبارتند از:

$$(۱, ۲, ۴, ۳) \quad , \quad \sum_i W_iT_i = ۴۸۰$$

$$(۲, ۴, ۱, ۳) \quad , \quad \sum_i W_iT_i = ۴۳۶ \quad \leftarrow$$

$$(۲, ۱, ۳, ۴) \quad , \quad \sum_i W_iT_i = ۶۵۲$$

همان‌طور که ملاحظه می‌شود در این تکرار، جواب $(۲, ۴, ۱, ۳)$ که تابع هدف کمتری از بین سایرین دارد انتخاب می‌شود و در مرحله بعد حرکت به آن صورت می‌گیرد. در این مثال، ما حرکت‌های صورت گرفته را در فهرست ممنوع قرار می‌دهیم. از آن‌جا که جابه‌جایی بین کارهای $j_۱$، $j_۴$ در $x_۱$ باعث به‌وجود آمدن $x_۲ = (۲, ۴, ۱, ۳)$ شده‌است، این حرکت را به شکل $(۱, ۴)$ نشان می‌دهیم و در تکرار بعدی آن را در فهرست ممنوع قرار می‌دهیم.

تکرار ۲: لیست ممنوع برابر با $TabuList = \{(۱, ۴)\}$ است.

$$x_۲ = (۲, ۴, ۱, ۳) \quad , \quad \sum_i W_iT_i = ۴۳۶$$

همسایگی‌های $x_۲$ به‌صورت زیر هستند:

$$(۴, ۲, ۱, ۳) \quad , \quad \sum_i W_iT_i = ۴۶۰ \leftarrow$$

$$(۲, ۱, ۴, ۳) \quad , \quad \sum_i W_iT_i = ۵۰۰ \quad \leftarrow Tabu$$

$$(۲, ۴, ۳, ۱) \quad , \quad \sum_i W_iT_i = ۶۰۸$$

در این مرحله مقدار تابع هدف تمام جواب‌های همسایه از مقدار تابع هدف $x_۲$ بدتر است، ولی باز هم به بهترین جواب همسایه یعنی $(۴, ۲, ۱, ۳)$ می‌رویم، چون ممکن است در دام یک جواب بهینه محلی افتاده باشیم، از طرفی شرط توقف نیز برقرار نشده است. با توجه به جواب انتخاب شده، در مرحله بعد، $(۲, ۴)$ به انتهای فهرست ممنوع اضافه می‌شود.

تکرار ۳: لیست ممنوع برابر با $TabuList = \{(۱, ۴), (۲, ۴)\}$ است.

$$x_۳ = (۴, ۲, ۱, ۳) \quad , \quad \sum_i = W_i T_i = ۴۶۰$$

همسایگی‌های $x_۳$ به‌صورت زیر می‌باشد:

$$(۲, ۴, ۱, ۳) \quad , \quad \sum_i W_i T_i = ۴۳۶ \leftarrow Tabu$$

$$(۴, ۱, ۲, ۳) \quad , \quad \sum_i W_i T_i = ۴۴۰ \quad \leftarrow$$

$$(۴, ۲, ۳, ۱) \quad , \quad \sum_i W_i T_i = ۴۳۶$$

در این مرحله جوابی که بهترین تابع هدف را داشت، انتخاب نشد. چون رسیدن به آن ملزم به انجام حرکتی بود که در فهرست ممنوع قرار داشت. بنابراین دومین جواب خوب همسایه انتخاب شد.

با انتخاب $(۴, ۱, ۲, ۳)$ باید حرکت $(۱, ۲)$ به فهرست ممنوع اضافه شود. از آن‌جا که طول فهرست ممنوع برابر با ۲ قرار داده شده‌است، حتماً باید یک حرکت از آن خارج شود که حرکت $(۱, ۴)$ از فهرست ممنوع خارج می‌شود.

تکرار ۴: لیست ممنوع برابر با $TabuList = \{(۲, ۴), (۱, ۲)\}$ است.

$$x_۴ = (۴, ۱, ۲, ۳) \quad , \quad \sum_i W_i T_i = ۴۴۰$$

همسایگی‌های $x_۴$ به‌صورت زیر می‌باشد:

$$(۱, ۴, ۲, ۳) \quad , \quad \sum_i W_i T_i = ۴۰۸ \leftarrow$$

$$(۴, ۲, ۱, ۳) \quad , \quad \sum_i W_i T_i = ۴۶۰$$

$$(۴, ۱, ۳, ۲) \quad , \quad \sum_i W_i T_i = ۵۸۶$$

همان‌طور که ملاحظه می‌شود در این مرحله شرط توقف برقرار می‌شود یعنی با انتخاب $(۱, ۴, ۲, ۳)$ مقدار تابع هدف برابر با ۴۰۸ می‌شود و بنابراین متوقف می‌شویم.

۵-۲-۲ استراتژی‌های پیشرفته‌ی الگوریتم جستجوی ممنوعه

ساختار کلی جستجوی ممنوعه اغلب جوابگوی مسائل بزرگ نیست. در واقع، روش جستجوی ممنوعه در متون علمی در دو نسخه‌ی به ظاهر متفاوت ارائه شده‌است که یکی ساده‌تر و دیگری پیشرفته‌تر خوانده می‌شود. نسخه ساده‌تر که بیانگر ساختار بیان شده‌است نسبت محدودی از مفاهیم روش جستجوی ممنوعه را به‌کار می‌گیرد و معمولاً برای تحلیل اولیه و تست عملکرد روش تنها از حافظه‌های کوتاه‌مدت استفاده می‌کند. در نسخه پیشرفته‌تر، عناصر دیگری به استراتژی جستجو اضافه می‌شود تا آن‌را اثربخش‌تر کند و در حقیقت از حافظه بلندمدت نیز استفاده می‌کند. در ادامه به بیان این عناصر خواهیم پرداخت.

۱. **لیست کاندید:** در یک جستجوی ممنوعه ساده، برای حرکت از جواب فعلی به جواب همسایه باید مقدار تابع هدف برای هر عنصر از $N(s)$ ارزیابی شود که این کار از لحاظ محاسباتی بسیار هزینه‌بر است. می‌توان به جای این‌که تمامی همسایه‌ها را بررسی کرد، تنها یک زیرمجموعه‌ی $N'(s)$ از $N(s)$ را در نظر گرفت که در نتیجه هزینه‌ی محاسباتی به‌طور قابل ملاحظه‌ای کاهش می‌یابد. استفاده از استراتژی لیست کاندید راهی است برای کنترل تعداد حرکت‌هایی که باید ارزیابی شوند. رویکردهای احتمالی، به‌عنوان یکی از استراتژی‌های لیست کاندید در نظر گرفته می‌شوند. اشتباه در طراحی یک لیست کاندید مناسب، یکی از عواملی است که منجر به نارسایی و پیاده‌سازی ضعیف TS می‌شود.

۲. **قوی‌سازی:** ایده‌ی اصلی در استراتژی قوی‌سازی این است که مناطقی از فضای جستجو که امید می‌رود جواب‌های بهتری در آن‌ها یافت شود، بهتر جستجو شود. بنابراین، در طول زمان جستجو می‌توان فرآیند جستجوی معمولی را متوقف و یک مرحله‌ی تقویت را اضافه نمود. این استراتژی مبتنی بر حافظه میان مدت است. دو رویکرد برای استراتژی قوی‌سازی عبارتند از:

- جستجو را از بهترین جوابی که تاکنون یافت شده دوباره شروع کنیم و اجزایی را که جذاب‌تر به‌نظر می‌رسند در آن ثابت نگه داریم.
- ساختار همسایگی را تغییر دهیم به‌گونه‌ای که به حرکت‌های قدرتمندتر مجوز اجرا داده شود.

۳. **متنوع‌سازی:** روش‌های مبتنی بر جستجوی محلی، آن‌قدر محلی هستند که زمان زیادی و یا تمامی زمان خود را در بخش محدودی از فضای جستجو صرف می‌کنند. در نتیجه، هر چند جواب‌های خوبی به‌وسیله‌ی این روش‌ها به‌دست می‌آیند، اما ممکن است جستجو از اکتشاف مناطق بهتر باز بماند و بنابراین به جواب‌هایی برسد که از جواب بهینه، بسیار دور هستند. متنوع‌سازی، یک مکانیزم الگوریتمیک است که برای حل این مشکل تلاش می‌کند. برای انجام این کار، متنوع‌سازی، جستجو را مجبور می‌کند به سوی مناطقی که تاکنون کشف نشده‌اند حرکت کند. این استراتژی معمولاً مبتنی بر حافظه بلندمدت است. دو رویکرد اصلی برای متنوع‌سازی وجود دارد:

- اجزایی که به ندرت یافت شده‌اند را در جواب فعلی (بهترین جواب شناسایی شده) قرار می‌دهیم و جستجو را از این نقطه مجدداً آغاز می‌کنیم که متنوع‌سازی شروع مجدد نامیده می‌شود.
- این رویکرد که متنوع‌سازی پیوسته نام دارد، استراتژی‌های متنوع‌سازی را در یک فرآیند جستجو، یکپارچه می‌کند. این کار را می‌توان با اضافه کردن مقدار جریمه به تابع هدف، برای اجزایی که به تناوب در حرکت‌های انجام شده وجود داشته‌اند، انجام داد.

۴. **مجوز دادن به جواب‌های نشدنی** توجه به تمامی محدودیت‌های مسأله در تعریف فضای جستجو، اغلب فرآیند جستجو را بسیار محدود می‌کند و می‌تواند منجر به جواب‌های نامطلوب شود. در این چنین حالت‌هایی، آزادسازی محدودیت‌ها یک استراتژی مناسب است ، زیرا فضای جستجوی بزرگتری را ایجاد می‌کند. برای این منظور، محدودیت مورد نظر را از تعریف فضای جستجو حذف کرده و یک مقدار جریمه به تابع هدف می‌افزاییم.

۲-۲-۶ جستجوی ممنوعه برای حل مسأله USAHLP

در مرجع [۲۶] الگوریتم جستجوی ممنوعه برای حل مسأله USAHLP پیشنهاد شده است. این مسأله در بخش ۱-۵ توصیف شد. از آن‌جایی که این مسأله متعلق به کلاس مسائل NP- سخت است، الگوریتم‌های فراابتکاری بسیاری برای حل آن پیشنهاد شده‌اند. الگوریتم جستجوی ممنوعه از مؤثرترین و پربازده‌ترین تکنیک‌ها برای حل مسائل بهینه‌سازی است. با توجه به برتری معنی‌دار جواب‌های بدست آمده توسط الگوریتم جستجوی ممنوعه در مقایسه با بهترین جواب‌هایی که قبلاً توسط رویکردهای دیگر پیداشده‌اند، این الگوریتم در سطح وسیعی از حوزه‌های تحقیقاتی به‌کار می‌رود. در ادامه برای حل مسأله USAHLP یک الگوریتم جستجوی ممنوعه ارائه می‌شود که از جستجوی ممنوعه پیشنهاد شده در [۲۷]،الهام گرفته شده است و مختصراً به‌صورت الگوریتم TSUSAHLP به آن ارجاع داده خواهدشد.

فرض کنید در فرمولبندی مسأله مکانیابی هاب تک تخصیصی بدون ظرفیت که قبلاً بیان شد، برای هر گره $i \in N$ که $N = \{۱, ۲, \cdots, n\}$ مجموعه گره‌ها است، $l(i)$ به عنوان هاب تخصیص داده شده به گره i معرفی شود. در این‌صورت، هزینه انتقال هر واحد جریان از گره i به گره j برابر است با

$$\chi C_{il(i)} + \alpha C_{l(i)l(j)} + \delta C_{l(j)j}$$

اگر $H \subseteq N$ بیانگر مجموعه هاب‌ها و $H' = N - H$ بیانگر مجموعه گره‌های غیرهاب و تابع $l : N \to H$ به‌عنوان تابع تخصیص در نظر گرفته شود، در این‌صورت، مقدار تابع هدف مربوط به (H, l) به‌صورت زیر بیان می‌گردد:

$$f(H,l) = \sum_{i\in H'}\sum_{j\in H'} W_{ij}(\chi C_{il(i)} + \alpha C_{l(i)l(j)} + \delta C_{l(j)j}) + \sum_{i\in H}\sum_{j\in H'} W_{ij}(\alpha C_{il(j)} + \delta C_{l(j)j})$$
$$+ \sum_{i\in H'}\sum_{j\in H} W_{ij}(\chi C_{il(i)} + \alpha C_{l(i)j}) + \sum_{i\in H}\sum_{j\in H} W_{ij}(\alpha C_{ij}) + \sum_{k\in H} F_k$$

(۱-۲)

بنابراین، (H^*, l^*) یک جواب بهینه است اگر

$$(H^*, l^*) = arg\, min_{H\subseteq N} f(H,l).$$

همان‌طور که در فصل اول بیان شد، سه نقش عمده تسهیلات هاب، اجرای سه فاز جمع‌آوری، انتقال و توزیع جریان است. در عبارت اول معادله ۲-۱، هر دو گره مبدأ و مقصد غیرهاب هستند. بنابراین، هر سه فاز قابل اجراست. از آنجایی که در عبارت دوم، مبدأ گره‌ای هاب و مقصد گره‌ای غیرهاب و همچنین در عبارت سوم، مبدأ گره‌ای غیرهاب و مقصد گره‌ای هاب است؛ لذا، در عبارت‌های دوم و سوم به ترتیب فازهای جمع‌آوری و توزیع اجرا نخواهد شد. در عبارت چهارم، هم گره مبدأ و هم گره مقصد هاب‌اند و از این‌رو فقط فاز انتقال اجرا می‌شود. نهایتاً، محاسبه مجموع هزینه‌ی تأسیس هاب‌ها برعهده‌ی عبارت پنجم است.

محاسبه مقدار تابع هدف، به‌عنوان زمان‌برترین بخش الگوریتم شناخته می‌شود. در الگوریتم TSUSAHLP ، تابع هدف فقط یک‌بار در ابتدای الگوریتم با استفاده از معادله ۲-۱ مورد محاسبه قرار می‌گیرد و بعد از آن مقدار تابع هدف با بکارگیری گزاره‌های زیر بروزرسانی خواهدشد. در نتیجه، زمان محاسباتی الگوریتم به‌طور قابل ملاحظه‌ای کاهش می‌یابد.

در الگوریتم مورد بحث، دو نوع حرکت بر روی گره‌ها انجام می‌شود:

- حرکت‌های مکانیابی: مجموعه هاب‌ها را تغییر می‌دهند؛

- حرکت پایه‌ای: شامل افزودن یا حذف یک گره هاب است و به‌صورت جابجایی از مجموعه H به مجموعه $\bar{H}$ در $N_1(H)$ بروز می‌کند.

$$N_1(H) = \{\bar{H} | \bar{H} = H \cup \{i\}, i \in H^{'}, or \bar{H} = H - \{i\}, i \in H\}$$

- حرکت معاوضه‌ای: یک هاب و یک غیرهاب تعویض می‌شوند که با حرکت از مجموعه H به مجموعه $\bar{H}$ در $N_2(H)$ انجام می‌گیرد.

$$N_2(H) = \{\bar{H} | \bar{H} = H \cup \{i\} - \{j\}, i \in H^{'}, j \in H\}$$

- حرکت تخصیص: تابع تخصیص l را اصلاح می‌کند.

انجام حرکت تخصیص روی گره غیرهاب r، تخصیص این گره را از $l(r)$ به $l^{'}(r)$ تغییر می‌دهد، به‌طوریکه به ازای هر $i \neq r$ داریم، $l^{'}(i) = l(i)$. تغییر در مقدار تابع هدف، از Z به $Z^{'}$ که به ترتیب به توابع تخصیص l و $l^{'}$ مربوط می‌شوند، با $\Delta Z^{r}_{l,l'}$ نشان داده می‌شود. گزاره ۲-۲-۱۰ نشان می‌دهد که $\Delta Z^{r}_{l,l'}$ می‌تواند در زمان $O(n)$ محاسبه شود.

گزاره ۲-۲-۱۰. برای هر $r \in H'$، مقدار $\Delta Z^{r}_{l,l'}$ به صورت زیر محاسبه می‌گردد:

$$\begin{aligned}\Delta Z^{r}_{l,l'} = \chi \sum_{i=1}^{n} W_{ri}(C_{rl'(r)} - C_{rl(r)}) + \alpha \sum_{i=1}^{n} W_{ir}(C_{l'(r)l(i)} - C_{l(r)l(i)}) \\ + \delta \sum_{i=1}^{n} W_{ir}(C_{l'(r)r} - C_{l(r)r}) + \alpha \sum_{i=1}^{n} W_{ir}(C_{l(i)l'(r)} - C_{l(i)l(r)}) \\ - \alpha W_{rr}(C_{l'(r)l(r)} - C_{l(r)l'(r)})\end{aligned} \tag{۲-۲}$$

برهان. جهت مطالعه برهان این گزاره به مرجع [۲۶] مراجعه کنید.

□

در حرکت مکانیابی پایه‌ای، وقتی گره هاب k از مجموعه هاب‌ها حذف می‌شود و یا به‌عبارت دیگر به یگ گره غیرهاب تبدیل می‌گردد، تمامی گره‌های تخصیص داده شده به گره k که خود گره k را نیز شامل می‌شوند به نزدیکترین هاب در $H - \{k\}$ تخصیص می‌یابند. فرض کنید

$$P = \{j | j \in H^{'} - \{k\}, L(j) = k\}$$

و قرار دهید $P = \{j_1, j_2, \cdots, j_m\}$ به قسمی که $j_1 < j_2 < \cdots < j_m$. گزاره ۲-۲-۱۱ نشان می‌دهد که در این حرکت مکانیابی، محاسبه ΔZ دارای مرتبه‌ی زمانی $O(mn)$ است که $m = |P|$.

گزاره ۲-۲-۱۱. اگر گره هاب k از مجموعه هاب‌ها حذف شود، آنگاه تغییر در مقدار تابع هدف به صورت زیر محاسبه می‌شود:

$$\Delta Z = -F(K) + \sum_{t=1}^{m} \Delta Z^{j_t}_{l'_{j_{t-1}}, l'_{j_t}} + \Delta Z^{k}_{l'_{j_m}, l'_k} \tag{۲-۳}$$

برهان. جهت مطالعه برهان این گزاره به مرجع [۲۶] مراجعه کنید. □

تبدیل گره غیرهاب به گره هاب، نوع دیگری از حرکت مکانیابی پایه‌ای است. فرض کنید که گره k به مجموعه‌ی گره‌های هاب اضافه شود. در این‌صورت، این گره باید به خودش اختصاص یابد و سایر گره‌های غیرهاب که به گره k نسبت به گره‌های

هاب قبلی‌شان نزدیکتر هستند، این گره را به‌عنوان هاب جدیدشان اختیار خواهند کرد. فرض کنید

$$Q = \{j | j \in H^{'} - \{k\}, C(j, l(j)) > C(j, k)\}.$$

مجموعه فوق را به‌صورت $Q = \{j_۱, j_۲, \cdots, j_m\}$ در نظر بگیرید که در آن رابطه $j_۱ < j_۲ < \cdots < j_m$ برقرار است. طبق گزاره ۲-۲-۱۲، در این نوع حرکت مکانیابی نیز محاسبه ΔZ از مرتبه زمانی $O(mn)$ است و $m = |Q|$.

گزاره ۲-۲-۱۲. فرض کنید که گره غیرهاب k به مجموعه هاب ها اضافه شود. در این حالت، تغییر در مقدار تابع هدف برابر است با

$$\Delta Z = F(K) + \Delta Z^{k}_{l,l^{'}_{k}} + \sum_{t=۱}^{m} \Delta Z^{j_t}_{l^{'}_{j_{t-۱}}, l^{'}_{j_t}} \qquad (۴-۲)$$

برهان. جهت مطالعه برهان این گزاره به مرجع [۲۶] مراجعه کنید. □

حافظه کوتاه‌مدت و حافظه بلندمدت که در الگوریتم TSUSAHLP مورد استفاده قرار می‌گیرند، یک چرخه را تشکیل می‌دهند. در آغاز هر چرخه، حافظه کوتاه‌مدت اجرا می‌شود. سپس، حافظه بلندمدت فراخوانی می‌گردد و جواب ایجاد شده در این حافظه به چرخه بعدی انتقال می‌یابد. در صورت برقراری شرط توقف، الگوریتم ۲-۵ اجرا شده و بهترین مقدار تابع هدف برگردانده می‌شود.

الگوریتم ۲-۵ روش TabuAlloc [۲]

ورودی: حداکثر تعداد تکرارها و بهترین جواب جاری.
خروجی: بهترین مقدار تابع هدف.

۱: مراحل زیر را تا زمانیکه حداکثر تعداد تکرارهای ممنوعه بدست آید انجام دهید:
۲: برای هر تخصیص مجدد قابل قبول یک گره غیرهاب،بهبود تابع هدف مبتنی بر کاهش هزینه را محاسبه کنید.
۳: بهترین تخصیص مجدد شدنی را اجرا کنید.
۴: اگر جواب بدست آمده باعث بهبود بهترین جواب جاری شود، آنگاه این جواب را به‌عنوان بهترین جواب جاری ثبت کنید.
۵: لیست ممنوعه تخصیص را بروزرسانی کنید.

فضای جواب الگوریتم TSUSAHLP به‌صورت زیر است:

$$A = \{(H, l) | l : N \to H, H \subseteq N, l(i) = i, \forall i \in H\}.$$

همان‌طور که می‌دانیم، الگوریتم جستجوی ممنوعه یک الگوریتم بهبوددهنده است نه سازنده. لذا باید به آن یک جواب

اولیه داده شود. با اتخاذ روش DROP [۲۷] برای حل مسأله USAHLP ، می‌توان مجموعه هاب‌های جواب اولیه را تعیین کرد. در روش DROP اتخاذ شده، در ابتدا همه‌ی گره‌ها هاب هستند (یعنی $S = N$). در صورتی که کاهش تابع هدف تعریف شده در ۲-۱ مربوط به گره j بیشتر از سایر گره‌های هاب باشد، این گره مجموعه هاب‌ها را ترک می‌کند و این مجموعه به‌صورت $H \leftarrow H - \{j\}$ بروزرسانی می‌گردد. این فرآیند تا زمانی تکرار می‌شود که دیگر بهبودی در مقدار تابع هدف دیده نشود. جهت تعیین تابع تخصیص هر مجموعه از هاب‌ها، قاعده فاصله محور استفاده خواهد شد. جزئیات ساختار جواب اولیه، در الگوریتم ۲-۶ آمده است. در این الگوریتم، ΔZ_j نشان دهنده‌ی تغییر در مقدار تابع هدف، زمان حذف گره j از مجموعه هاب‌هاست که توسط گزاره ۲-۲-۱۱ محاسبه می‌شود.

الگوریتم ۲-۶ ایجاد جواب اولیه با روش DROP

Require: N, Parameter α, matrices C_{ij}, W_{ij} and F_i.
Ensure: H, H' and l.
  while $p = 1$ or $\exists j \in N; \Delta z_j < 0$ **do**
    $\Delta z_{j^*} \leftarrow min\{\Delta z_j | j \in H, \Delta z_j < 0\}$
    $H \leftarrow H - \{j^*\}$
    $z \leftarrow z + \Delta z_{j^*}$
  end while
  $H' = N - H$
  for all $j \in H'$ **do**
    $i^* = min\{i | R_{ij} \in H\}$
    $l(j) = R_{i^*j}$
  end for

در الگوریتم، گاهی اوقات لازم است که نزدیکترین هاب به هر گره مشخص شود. بدین منظور، ماتریس معین R معرفی می‌گردد که در آن ستون j شامل اندیس گره‌هایی است که فاصله‌شان از این گره به‌صورت صعودی مرتب شده‌اند. از آنجایی که هر گره نزدیکترین گره به خودش است؛ واضح است که برای هر $j \in N$ داریم، $R_{1j} = j$. با استفاده از ماتریس معین R، محاسبه و بروزرسانی تابع $f(.,.)$ تعریف شده در ۲-۱ خیلی سریعتر انجام خواهد گرفت.

جهت انتخاب حرکت مکانیابی برای مجموعه‌ای از هاب‌ها، بایستی همسایگی‌های H مشخص شوند. توجه داشته باشید که یافتن تخصیص مناسب برای هر مجموعه از هاب‌ها بسیار زمانبر است. لذا، تخصیص شدنی خوب براساس تخصیص قبلی و با استفاده از ماتریس معین R انجام می‌گیرد. اگر گره هاب i مجموعه جاری هاب‌ها را ترک کند (یعنی $\bar{H} = H - \{i\}$)، آنگاه گره‌هایی که به هاب i تخصیص داشته‌اند به نزدیکترین هاب در $\bar{H}$ تخصیص خواهند یافت. در غیر این‌صورت، اگر گره i به مجموعه جاری هاب‌ها اضافه شود (یعنی $\bar{H} = H \cup \{i\}$) گره‌هایی که به این گره نسبت به هاب‌های خودشان نزدیکترند، به هاب جدید i اختصاص می‌یابند و بروزرسانی مقدار تابع هدف، به ترتیب با استفاده از گزاره‌های ۲-۲-۱۱ و ۲-۲-۱۲ انجام خواهد گرفت. نهایتاً برای انجام حرکت معاوضه‌ای، هابی از مجموعه هاب‌ها حذف

می‌شود و غیرهابی به این مجموعه اضافه می‌گردد. بنابراین، برای این حرکت دو فرآیند بالا به ترتیب انجام می‌شوند.

قابل توجه است که وقتی یک حرکت مکانیابی انجام می‌گیرد، هر گره غیرهاب به گره‌های هاب اختصاص داده خواهد شد. فرض کنید (H, l) جواب جاری باشد. تعیین تابع تخصیص برای مجموعه هاب‌ها به طریق زیر است:

$$l^*(j) = arg\,min\{f(H,l)|l(j) = i, i \in H\} \quad \forall j \in H' \qquad (۵-۲)$$

همان‌طور که قبلاً نیز بیان شد، اجرای حافظه کوتاه مدت در آغاز هر چرخه الگوریتم صورت می‌گیرد. هدف حافظه کوتاه مدت، یافتن جواب‌های جدیدی است که در حرکت‌های اخیر جستجو دیده نشده‌اند. بنابراین، وقتی‌که وضعیت گره‌ای تغییر می‌کند، این گره برای یک دوره ممنوعیت در لیست ممنوعه قرار داده می‌شود. برای این‌که از تغییرمکان‌های نامناسب در طی الگوریتم جلوگیری به‌عمل آید، از لیست کاندید استفاده خواهدشد. با تعریف

$$d = \frac{y}{n^۲} \sum_{i=۱}^{n} \sum_{j=۱}^{n} C_{ij} \qquad (۶-۲)$$

که در آن $y \in R^+$ پارامتری از پیش تعریف شده است، اگر فاصله گره غیرهاب j تا نزدیکترین هابش کمتر از d باشد، این گره علامت‌گذاری می‌شود. لیست کاندید شامل تمامی گره‌های بدون علامت است. این نوع لیست کاندید قبلاً در هیچ جستجوی ممنوعه دیگری که برای مسائل مکانیابی ارائه شده‌اند، استفاده نشده‌است. هنگامی‌که موقعیت گره‌ای تغییر می‌یابد، آن گره به لیست ممنوعه برای یک دوره ممنوعیت اضافه می‌شود. مطالعه محاسباتی نشان داده است که دوره ممنوعیت ثابت یا تصادفی استفاده شده در الگوریتم سان، زمان محاسباتی را کاهش می‌دهد اما معمولاً کیفیت جواب پایین می‌آید. بنابراین، در اینجا روشی تطبیقی برای دوره ممنوعیت به‌کار می‌رود که تمایل به حفظ گره‌هایی با بیشترین (یا کمترین) هزینه و جریان ارسالی به‌عنوان هاب (یا غیرهاب) برای مدتی طولانی دارد. فرض کنید

$$A(i) = \frac{\sum_{j=۱}^{n}(W_{ij}C_{ij} + W_{ji}C_{ji})}{\sum_{k=۱}^{n}\sum_{j=۱}^{n}(W_{kj}C_{kj} + W_{jk}C_{jk})} \quad \forall i \in N.$$

بردارهای l_c وl_o به ترتیب بیانگر دوره ممنوعیت گره‌های غیرهاب و گره‌های هاب هستند که $l_c(i)$ و $l_o(i)$ برای $i \in N$ به‌صورت زیر محاسبه می‌شوند:

$$l_c(i) = min\{(۱ - A(i))(nx_۱) + ۱, nx_۲\} \quad \forall i \in N \qquad (۷-۲)$$

$$l_o(i) = min\{A(i)(n\sqrt{n}x_۳) + ۱, nx_۴\} \quad \forall i \in N \qquad (۸-۲)$$

که $x_۱, x_۲ \in R^+$ و همچنین $x_۳$ و $x_۴$ پارامترهای از پیش تعریف شده‌اند.

آرایه $t \subseteq Z$ برای اندازه‌گیری تازگی یک جواب استفاده می‌شود. عنصر t_j بیان کننده تکراری است که در آن وضعیت گره j تغییر داده شده‌است که $j = ۱, ۲, \cdots, n$. مقدار اولیه t براساس مقادیر l_c و l_o به‌صورت زیر مشخص می‌گردد:

$$t(j) = \begin{cases} -l_o(j) & l(j) = j \\ l_c(j) & l(j) \neq j \end{cases} \quad (۹-۲)$$

بروزرسانی t_j همزمان با تغییر موقعیت گره j اتفاق می‌افتد. در تکرار k ام گره j در لیست ممنوعه قرار دارد اگر

$$\begin{cases} k - t(j) < l_o(j) & l(j) = j \\ k - t(j) < l_c(j) & l(j) \neq j \end{cases}$$

به بیان دیگر، هنگامی‌که وضعیت گره j به هاب تغییر می‌کند، این گره بایستی برای $l_o(j)$ تکرار، هاب باقی بماند. به‌طور مشابه، وقتی وضعیت گره j به غیرهاب تغییر می‌یابد، باید برای $l_o(j)$ تکرار، به‌عنوان یک گره غیرهاب عمل کند.

گاهی اوقات، جواب‌های خوبی که هیچ‌گونه خطری برای دوری شدن الگوریتم ایجاد نمی‌کنند با قوانین ممنوعه قوی محدود می‌شوند. برای رفع این ممنوعیت‌ها، ضابطه تنفس به‌کار می‌رود. فرض کنید z مقدار جاری تابع هدف باشد. با توجه به این‌که

- Δz_j: تغییر در مقدار تابع هدف، هنگام تغییر وضعیت گره j در حرکت پایه‌ای؛
- Δz_{ij}: تغییر در مقدار تابع هدف، هنگام تعویض گره هاب i و گره غیرهاب j در حرکت معاوضه‌ای؛
- z_*: بهترین مقدار تابع هدف پیدا شده در چرخه جاری؛
- z_{**}: بهترین مقدار تابع هدف بدست آمده؛

در صورتی ضابطه تنفس در حرکت پایه‌ای برای گره j صادق است که

$$z + \Delta z_j < z_{**}$$

و همچنین، در حرکت معاوضه‌ای، ضابطه تنفس برای گره‌های i و j برقرار است اگر

$$z + \Delta z_{ij} < z_{**}$$

برای انجام یک حرکت پایه‌ای، لیست کاندید (مجموعه‌ی گره‌های بدون علامت) ساخته می‌شود. فرض کنید

$$j^* = arg\,min\{\Delta z_j | j \in N, j \text{ is unflagged}\}$$

اگر گره j^* در لیست ممنوعه نباشد یا اگر هست ضابطه تنفس برای آن صادق باشد، حرکت پایه‌ای انجام می‌گیرد. در غیر این‌صورت، به منظور اجتناب از بهاگذاری مجدد گره j^* در تکرار جاری، این گره علامت‌گذاری خواهدشد. این فرآیند تا زمانی ادامه می‌یابد که یک حرکت انجام گیرد. بعد از انجام یک حرکت، تمامی گره‌ها بدون علامت می‌گردند.

برای انجام یک حرکت معاوضه‌ای، (i^*, j^*) به طریق زیر مشخص می‌شوند:

$$(i^*, j^*) = arg\, min\, \Delta z_{ij}.$$

گره هاب i^* و گره غیرهاب j^* در صورتی می‌توانند تعویض شوند که در لیست ممنوعه نباشند و یا ضابطه تنفس برای آن‌ها برقرار باشد.

حرکت پایه‌ای در حافظه کوتاه مدت در صورت برقراری رابطه زیر متوقف می‌شود:

$$k - k_0 > n_1.$$

در رابطه فوق، n_1 پارامتری از پیش تعریف شده، k تکرار جاری و k_0 آخرین تکراری است که در آن z_0 بروزرسانی شده‌است. به‌عبارت دیگر، اگر بهترین جواب چرخه جاری در n_1 تکرار گذشته بهبود نیافته است، حرکت پایه‌ای متوقف خواهدشد. بعد از آن، یک حرکت معاوضه‌ای انجام می‌شود و سپس حافظه کوتاه مدت متوقف می‌گردد.

در حافظه بلند مدت، جواب‌ها با تغییر وضعیت گره‌ای که وضعیت خود را برای مدت طولانی تری در میان گره‌ها تغییر نداده‌است، متنوع می‌شوند. فرض کنید

$$j^* = arg \min_{j \in N} t(j).$$

اگر j^* تنها گره هاب موجود نباشد، وضعیت آن تغییر می‌کند و سپس تخصیص این جواب براساس ۲-۵ بهبود می‌یابد. این روند n_2 بار تکرار می‌شود که

$$n_2 = max\{1, [bn]\}.$$

در رابطه فوق، n بیانگر تعداد گره‌ها و b یک پارامتر از پیش تعیین شده‌است. پس از اینکه چرخه K بار تکرارشد (K پارامتری از پیش تعریف شده‌است)، الگوریتم متوقف شده و بهترین جواب پیداشده الگوریتم بازگردانده می‌شود. الگوریتم TSUSAHLP در الگوریتم ۲-۷ بیان شده‌است.

الگوریتم ۲-۷ الگوریتم TSUSAHLP

ورودی: پارامترهای $\mathbf{K}$ و $\mathbf{y}$ و $\mathbf{x}_1$ و $\mathbf{x}_2$ و $\mathbf{x}_3$ و $\mathbf{x}_4$ و $\mathbf{n}_1$ و $\mathbf{b}$،
خروجی: بهترین مقدار تابع هدف.

۱: ماتریس معین $\mathbf{R}$ را تشکیل دهید.
۲: فرض کنید $\mathbf{kk} \leftarrow 0$ (شمارنده تعداد چرخه‌ها) و $\mathbf{kl} \leftarrow 0$ (شمارنده تعداد تکرارها در حافظه بلند مدت) و $\mathbf{UF} \leftarrow \mathbf{N}$.
۳: $\mathbf{H}$ و $\mathbf{l}$ را با استفاده از الگوریتم ۲-۶ بسازید و فرض کنید $\mathbf{z} \leftarrow \mathbf{f}(\mathbf{H}, \mathbf{l})$ که $\mathbf{f}(.,.)$ در معادله ۲-۱ معرفی شده‌است.
۴: قرار دهید $\mathbf{z}_0 \leftarrow \mathbf{z}$ و $\mathbf{z}_{00} \leftarrow \mathbf{z}$ و $\mathbf{k} \leftarrow 0$ و $\mathbf{k}_0 \leftarrow 0$.
۵: بردارهای $\mathbf{l}_c$ و $\mathbf{l}_o$ را به ترتیب براساس معادله‌های ۲-۷ و ۲-۸ تشکیل دهید.
۶: بردار $\mathbf{t}$ تعریف شده در معادله ۲-۹ را بسازید.
۷: **{شروع حافظه کوتاه مدت}** فرض کنید $\mathbf{UF} \leftarrow \mathbf{N} - \{\mathbf{j} | \mathbf{j} \in \mathbf{H}', |\mathbf{j} - \mathbf{l}(\mathbf{j})| < \mathbf{d}\}$ که $\mathbf{d}$ در معادله ۲-۶ معرفی شده‌است.
۸: فرض کنید $\mathbf{j}^* \leftarrow \arg\min_{\mathbf{j} \in \mathbf{UF}} \Delta \mathbf{z}_\mathbf{j}$
اگر $\mathbf{k} - \mathbf{t}_{\mathbf{j}^*} \leq \mathbf{l}(\mathbf{j}^*)$ آنگاه $\mathbf{j}^*$ در لیست ممنوعه قرار دارد و به مرحله ۹ بروید؛
در غیر این‌صورت، فرض کنید $\mathbf{basic\ move} = 1$ و به مرحله ۱۰ بروید.
۹: اگر $\mathbf{z} + \Delta \mathbf{z}_{\mathbf{j}^*} < \mathbf{z}_{00}$ (ضابطه تنفس برقرار است)، آنگاه فرض کنید $\mathbf{basic\ move} = 1$ و به مرحله ۱۰ بروید؛
در غیر این‌صورت، فرض کنید $\mathbf{UF} \leftarrow \mathbf{UF} - \{\mathbf{j}^*\}$ و به مرحله ۸ بروید.
۱۰: **{بروزرسانی جواب جاری}** اگر $\mathbf{j}^* \in \mathbf{H}$ آنگاه $\mathbf{H} \leftarrow \mathbf{H} - \{\mathbf{j}^*\}$ و $\mathbf{p} \leftarrow \mathbf{p} - 1$؛
در غیر این‌صورت، $\mathbf{H} \leftarrow \mathbf{H} \cup \{\mathbf{j}^*\}$ و $\mathbf{p} \leftarrow \mathbf{p} + 1$.
۱۱: تابع تخصیص را براساس قاعده فاصله محور [۸] تشکیل دهید.
۱۲: فرض کنید $\mathbf{z} \leftarrow \mathbf{z} + \Delta \mathbf{z}_{\mathbf{j}^*}$ و $\mathbf{k} \leftarrow \mathbf{k} + 1$ و $\mathbf{t}(\mathbf{j}^*) \leftarrow \mathbf{k}$
۱۳: اگر $\mathbf{z} < \mathbf{z}_0$ آنگاه $\mathbf{z}_0 \leftarrow \mathbf{z}$ و $\mathbf{k}_0 \leftarrow \mathbf{k}$.
۱۴: اگر $\mathbf{z}_0 < \mathbf{z}_{00}$ آنگاه $\mathbf{z}_{00} \leftarrow \mathbf{z}_0$.
۱۵: اگر $\mathbf{swap\ move} = 1$ آنگاه قرار دهید $\mathbf{swap\ move} = 0$ و $\mathbf{j}^* \leftarrow \mathbf{i}^*$ و به مرحله ۱۰ بروید.
۱۶: اگر $\mathbf{k} - \mathbf{k}_0 < \mathbf{n}_1$ و $\mathbf{basic\ move} = 1$ ، آنگاه به مرحله ۷ بروید.
در غیر این‌صورت، اگر $\mathbf{k} - \mathbf{k}_0 = \mathbf{n}_1$ فرض کنید $\mathbf{basic\ move} = 0$ و در مرحله بعد حرکت معاوضه‌ای انجام می‌گیرد.
۱۷: فرض کنید $(\mathbf{i}^*, \mathbf{j}^*) = \arg\min \Delta \mathbf{z}_{\mathbf{i},\mathbf{j}}$
اگر $\mathbf{k} - \mathbf{t}_{\mathbf{i}^*} > \mathbf{l}(\mathbf{i}^*)$ و $\mathbf{k} - \mathbf{t}_{\mathbf{j}^*} > \mathbf{l}(\mathbf{j}^*)$ (یعنی $\mathbf{i}^*$ و $\mathbf{j}^*$ در لیست ممنوعه نیستند) یا $\mathbf{z} + \Delta \mathbf{z}_{\mathbf{i}^*,\mathbf{j}^*} < \mathbf{z}_{00}$ (یعنی ضابطه تنفس برقرار است) آنگاه فرض کنید $\mathbf{swap\ move} = 1$ و به مرحله ۱۰ بروید.
در غیر این‌صورت، اگر $\mathbf{kl} \leq \mathbf{n}_2$ به مرحله ۱۹ بروید و در غیر این‌صورت قرار دهید $\mathbf{kk} \leftarrow \mathbf{kk} + 1$.
۱۸: اگر $\mathbf{kk} = \mathbf{K}$ متوقف شوید. الگوریتم ۲-۵ را اجرا کنید و $\mathbf{z}_{00}$ را به‌عنوان بهترین مقدار تابع هدف برگردانید
در غبر این‌صورت، فرض کنید $\mathbf{k} \leftarrow \mathbf{k}_0$ و $\mathbf{kl} \leftarrow 0$ و $\mathbf{z} \leftarrow \mathbf{z}_0$ و به مرحله ۷ بروید.
۱۹: **{شروع حافظه بلندمدت}** قرار دهید $\mathbf{kl} \leftarrow \mathbf{kl} + 1$ و $\mathbf{UF} \leftarrow \mathbf{N}$.
۲۰: اگر $\mathbf{p} > 1$ یا $\mathbf{j}^*$ هاب نبود، آنگاه به مرحله ۱۰ بروید.
در غیر این‌صورت، به مرحله ۱۶ بروید.

۲-۳ الگوریتم ژنتیک (GA)

۲-۳-۱ مفاهیم پایه‌ای

در ادبیات، الگوریتم‌های ژنتیک برای مجموعه عظیمی از مسائل با زمینه‌های مختلف اعمال شده‌اند و به‌دلیل قدرت قابل توجه این الگوریتم در امر بهینه‌سازی، به یکی از پرکاربردترین روش‌های فراابتکاری برای حل مسائل بهینه‌سازی تبدیل شده‌است. الگوریتم ژنتیک، یکی از الگوریتم‌های تکاملی است. مفهوم الگوریتم تکاملی در زیربخش ۲-۱-۴-۲ ارائه شد. در ادامه به تشریح الگوریتم ژنتیک که معروف‌ترین الگوریتم تکاملی است می‌پردازیم.

تعریف ۲-۳-۱. هر جواب مسأله بهینه‌سازی یا به‌عبارت دیگر هر بردار از فضای حالت مسأله، بسته به نوع مسأله می‌تواند به‌صورت یک رشته از اعداد، یک ماتریس و ... نمایش داده شود که در این صورت به آن یک کروموزوم می‌گویند. برای نمایش کروموزوم‌ها، معمولاً از کدگذاری‌های دودویی (رشته‌های بیتی) استفاده می‌شود.

تعریف ۲-۳-۲. اعضای تشکیل دهنده‌ی کروموزوم، ژن نامیده می‌شوند. هر ژن مقدار یکی از متغیرهای مسأله را بیان می‌کند.

تعریف ۲-۳-۳. به مجموعه‌ای از کروموزوم‌ها که در کنار هم قرار گرفته‌اند، جمعیت گفته می‌شود. هر یک از کروموزوم‌ها به مثابه یکی از افراد جامعه است.

نقش جمعیت، نگه‌داری جواب‌های کاندید است. افراد، عناصر ثابتی هستند که تغییر یا بهبود نمی‌یابند و این جمعیت است که تغییر کرده و بهبود می‌یابد. تقریباً در تمامی کاربردهای الگوریتم‌های ژنتیک، اندازهِ جمعیت ثابت است و در طول جستجوی تکاملی تغییر نمی‌کند. تنوع [۱] یک جمعیت، معیار وجود تعداد جواب‌های متفاوت در آن جمعیت است. معیار منحصر بفردی برای تنوع وجود ندارد. معمولاً تعداد مقادیر برازندگی‌های متفاوت، تعدا کروموزوم‌های متفاوت و یا تعداد ژن‌های متفاوت برای تنوع در نظر گرفته می‌شوند.

- همگرایی زودرس، به معنای از دست رفتن تنوع جمعیت و قرار گرفتن الگوریتم در بهینه‌ی محلی است. این خطر در اکثر الگوریتم‌های تکاملی وجود دارد.

تعریف ۲-۳-۴. تابع برازندگی تابعی نامنفی است که میزان سازگار بودن یک کروموزوم را نسبت به هدف مسأله می‌سنجد.

در الگوریتم ژنتیک به دنبال ماکزیمم کردن مقدار برازندگی کروموزوم‌ها هستیم. با توجه به این‌که تابع برازندگی باید از نوع ماکزیمم‌سازی تعریف شود، در مسائلی که تابع هدف مسئله از نوع ماکزیمم‌سازی است، تابع برازندگی را می‌توان همان تابع هدف در نظر گرفت. اما در مسائلی که تابع هدف مسئله از نوع مینیمم‌سازی است، می‌توان با تبدیلاتی ساده نوع تابع هدف را به ماکزیمم‌سازی تبدیل و از تابع جدید به‌عنوان تابع برازندگی استفاده کرد.

---

[۱] diversity

تعریف ۲-۳-۵. هر تکرار الگوریتم را یک نسل می‌نامند.

یک الگوریتم تکاملی در طی نسل های متوالی، به‌طور تدریجی تکامل پیدا می‌کند. هدفی که در طول این نسل‌ها دنبال می‌شود، بهبود برازندگی افراد است. این بهبود، توسط شبیه‌سازی دو مکانیزم اصلی که در تکامل موجودات زنده براساس نظریه‌ی داروین وجود دارد، حاصل می‌شود:

- انتخاب: تضمین می‌کند افرادی که برازندگی بالاتری دارند، قابلیت تولید مثل بالاتر و بقای بیشتر دارند.
- تولیدمثل: اجازه‌ی ترکیب، تکثیر و تنوع ویژگی‌های والدین را می‌دهد تا فرزندانی با قابلیت‌های جدید تولید کنند.

حرکت از هر نسل به نسل بعدی، در چهار فاز صورت می‌گیرد:

۱. **فاز انتخاب**: افرادی را که در تولید مثل شرکت می‌کنند، تعیین می‌کند.

۲. **فاز تغییر**: شامل به‌کار بردن عملگرهای تغییر بر روی افراد انتخاب شده، برای ایجاد افراد جدید است.

۳. **فاز ارزیابی**: میزان برازندگی افراد جدید را ارزیابی می‌کند.

۴. **فاز جایگزینی**: شامل انتخاب اعضای نسل جدید است.

با توجه به مطالبی که بیان شد، در جدول ۲-۴ مقایسه‌ای بین مفاهیم بهینه‌سازی و فرآیند الگوریتم ژنتیک صورت گرفته است.

جدول ۲-۳: مقایسه‌ی مفاهیم بهینه‌سازی و فرایند الگوریتم ژنتیک

مسأله بهینه‌سازی ترکیبی	فرآیند تکامل تدریجی در الگوریتم ژنتیک
بردار جواب	کروموزم
عناصر بردار جواب	ژن‌ها
مقدار هر یک از عناصر بردار جواب	الل
مجموعه‌ای از بردارهای جواب	جمعیت
جواب‌های شدنی	فضای حالت
هر تکرار از الگوریتم	نسل
کیفیت حل برحسب مقدار تابع هدف	میزان سازگاری

۲-۳-۲ استراتژی‌های انتخاب

در چرخه‌ی تکامل الگوریتم ژنتیک، دو نقطه وجود دارد که در آن‌ها رقابت مبتنی بر برازندگی رخ می‌دهد:

- انتخاب افراد برای شرکت در تولید مثل (انتخاب والد)
- انتخاب افراد برای بقا در نسل بعدی (انتخاب بازمانده)

تعریف ۲-۳-۶. والد به جوابی گفته می‌شود که به منظور خلق جواب جدید (فرزند)، از میان افراد انتخاب می‌شود.

عملگرهای انتخاب (انتخاب والد و انتخاب بازمانده) تمامی جمعیت فعلی را مورد توجه قرار می‌دهند و به‌عبارت دیگر روی جمعیت کار می‌کنند. در الگوریتم ژنتیک، فرآیند انتخاب والد، احتمالی است. نقش انتخاب والد، تمایز قائل شدن بین افراد براساس برازندگی است. یعنی اجازه دادن به افراد بهتر برای این‌که والدین نسل بعد باشند. فرآیند انتخاب، کروموزم‌های والد را در یک جمعیت جدید آزمایشی که به‌عنوان استخر جفت‌گیری شناخته می‌شود، برای انجام عملیات ژنتیکی کپی می‌کند. تعداد نسخه‌هایی که یک فرد برای نسل بعدی دریافت می‌کند، معمولاً به‌طور مستقیم متناسب با مقدار برازندگی آن است و از این‌رو تا حدی از روش انتخاب طبیعی تقلید می‌کند. بنابراین افراد با برازندگی بالاتر شانس بیشتری برای والد شدن نسبت به افراد با برازندگی پایین‌تر دارند. با این وجود، افراد با برازندگی پایین، دارای شانس کوچک اما مثبت هستند. روش‌های انتخاب والد را به طور گسترده می‌توان به دو دسته زیر طبقه‌بندی کرد [۲۸]:

۲-۳-۲-۱ انتخاب مبتنی بر تناسب برازندگی

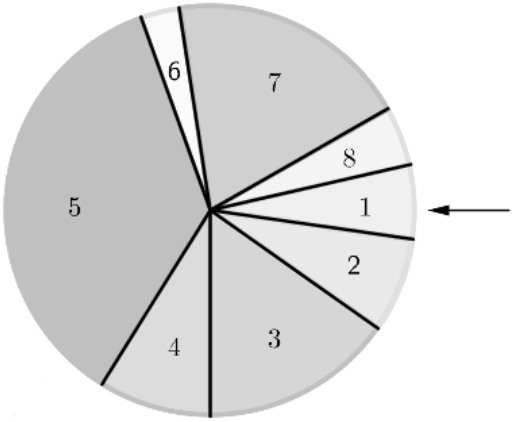
این استراتژی شامل روش‌هایی مانند انتخاب چرخ رولت [۱] و انتخاب جامع احتمالی [۲] است. در انتخاب چرخ رولت، متناسب با مقدار برازندگی هر فرد در جمعیت، قطاعی از چرخ رولت به آن تخصیص می‌یابد. این بدان معناست که در چرخ رولت پیشقدردار، جواب‌های خوب دارای اندازه قطاع بزرگتری نسبت به جواب‌های با کیفیت پایین‌تر هستند. بنابراین، تعداد دفعاتی که یک فرد انتخاب خواهد شد متناسب با مقدار برازندگی یا اندازه قطاع آن است. چرخ رولت برای تولید نامزد تولیدمثل دوران داده شده و پس از توقف، هر کروموزومی که در محل انتخاب مشخص شده بر روی چرخ قرار داشت، انتخاب می‌شود. در شکل ۲-۷ نمایی از استراتژی انتخاب مبتنی بر چرخ رولت برای جمعیتی با اندازه $n = ۸$ به تصویر کشیده شده‌است. طرح کلی انتخاب چرخ رولت با استفاده از الگوریتم ۲-۸ قابل اجرا است.

مثال ۲-۳-۷. جمعیتی با پنج نفر ($n = ۵$) را در نظر بگیرید. مقدار برازندگی این افراد در جدول ۲-۴ نشان داده شده‌است. مجموع برازندگی‌ها برابراست با:

$$\sum_{j=۱}^{n} f_j = ۲۸ + ۱۸ + ۱۴ + ۹ + ۲۶ = ۹۵.$$

با بکارگیری الگوریتم ۲-۸، احتمال انتخاب هر فرد و همچنین احتمال تجمعی متناظر نیز قابل محاسبه است که در این جدول نشان داده شده‌اند. اگر عدد تصادفی $r = ۰/۵۸۵$ را تولید کنیم، از آنجایی که $q_۲ = ۰/۴۸۴ < ۰/۵۸۵ \leq q_۳ = ۰/۶۳۱$، کروموزوم سوم انتخاب می‌شود.

[۱] roulette-wheel selection [۲] stochastic universal selection



شکل ۲-۷: انتخاب چرخ رولت

الگوریتم ۲-۸ شبه کد الگوریتم چرخ رولت

ورودی: تابع برازندگی f و اندازه جمعیت (n)

خروجی: تعیین تعداد حضور هر فرد به‌عنوان والد.

۱: مقدار برازندگی هر فرد در جمعیت، یعنی f_i را ارزیابی کنید.

۲: احتمال انتخاب هر عضو جمعیت (اندازه قطاع) را محاسبه کنید: $p_i = f_i / \sum_{j=1}^{n} f_j$

۳: احتمال تجمعی هر فرد را برآورد کنید: $q_i = \sum_{j=1}^{n} p_j$

۴: عدد تصادفی یکنواخت r را از بازه $[0, 1)$ تولید کنید.

۵: اگر $r < q_1$، آنگاه اولین کروموزوم (x_1) را انتخاب کنید. در غیر این‌صورت فرد x_i را بقسمی که $q_{i-1} < r \leq q_i$ انتخاب نمایید.

۶: مراحل ۴ تا ۵ را n بار برای تولید n کاندید جفت‌گیری تکرار کنید.

در انتخاب جامع احتمالی، P نشانگر متساوی الفاصله بر روی چرخ قرار می‌گیرند. کلیه P فرد موردنیاز با یک بار دوران چرخ انتخاب می‌شوند. تعداد نسخه‌هایی که از یک فرد بدست می‌آید برابر با تعداد نشانگرهایی است که در قطاع مربوطه قرار دارند. در شکل ۲-۸ این استراتژی انتخاب نشان داده شده‌است.

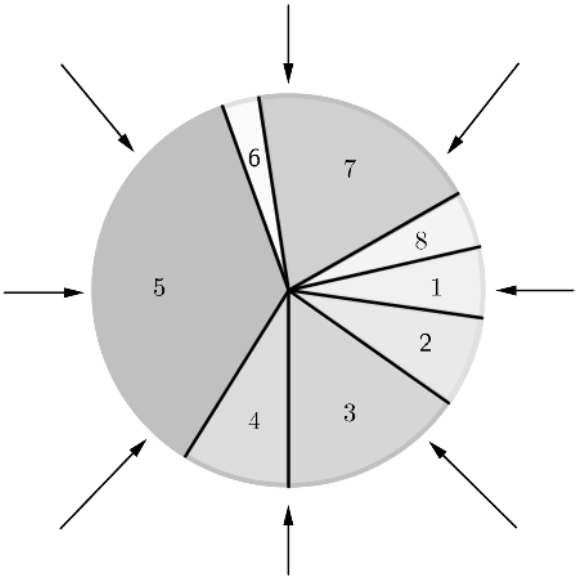
۲-۲-۳-۲ انتخاب وصفی

انتخاب وصفی، شامل روش‌هایی مانند انتخاب مسابقه‌ای [۱] و انتخاب برشی [۲] است. انتخاب مسابقه‌ای یکی از محبوب‌ترین روش‌های انتخاب است که در آن s کروموزوم به‌طور تصادفی انتخاب می‌شوند و با ورود به یک مسابقه در برابر یکدیگر قرار می‌گیرند. بهترین فرد از لحاظ برازندگی در گروه کروموزوم‌ها برنده مسابقه است و به‌عنوان والد انتخاب می‌گردد. اکثر اوقات، تعداد کروموزوم‌های شرکت کننده در مسابقه و یا به‌عبارت دیگر مقدار s برابر ۲ است. یکی از ویژگی‌های مثبت انتخاب مسابقه‌ای این است که نیاز به هیچ اطلاعات سراسری از جمعیت ندارد. لذا، از لحاظ مفهومی ساده و برای پیاده‌سازی،

[۱] tournament selection [۲] truncation selection

جدول ۲-۴: مقادیر برازندگی جمعیتی با اندازه $n = ۵$

کروموزوم‌ها	برازندگی (f)	احتمال (p_i)	احتمال تجمعی (q_i)
۱	۲۸	۰.۲۹۵	۰.۲۹۵
۲	۱۸	۰.۱۸۹	۰.۴۸۴
۳	۱۴	۰.۱۴۷	۰.۶۳۱
۴	۹	۰.۰۹۵	۰.۷۲۶
۵	۲۶	۰.۲۷۴	۱.۰۰۰



شکل ۲-۸: انتخاب جامع احتمالی

سریع عمل می‌کند به‌طوریکه تعداد n مسابقه برای انتخاب n فرد لازم است.

در انتخاب برشی، جواب‌های کاندید براساس مقدار برازندگی‌شان مرتب می‌شوند و نسبتی (مثلا $p = ۱/۲, ۱/۳, \ldots$) از افراد مناسب‌تر انتخاب شده و $۱/p$ ام بار تکثیر می‌شوند. انتخاب برشی از بسیاری از روش‌های انتخاب دیگر پیچیده‌تر است و اغلب در عمل کاربرد ندارد.

۲-۳-۳ عملگرهای تغییر

عملگرهای تغییر، تنوع موردنیاز را در الگوریتم ژنتیک ایجاد می‌کنند و سبب بهبود مقادیر برازندگی در نسل‌های متوالی می‌شوند. این عملگرها، گام‌های نخست فرآیند تکاملی را در فضای جستجو تشکیل می‌دهند. تولید یک فرزند، به معنای گام برداشتن به سمت یک نقطه‌ی جدید در این فضاست. در الگوریتم ژنتیک دو عملگر تغییر به‌کار برده می‌شوند که در ادامه به تشریح این عملگرها می‌پردازیم. یکی از تفاوت‌های عمده این عملگرها در آن است که عمل تقاطع به حداقل دو

والد نیاز دارد، اما در هنگام جهش به یک والد نیازمندیم.

۲-۳-۳-۱ عملگر تقاطع

تعریف ۲-۳-۸. به ترکیب ژنتیکی دو یا چند کروموزم به منظور تولید فرزند، عمل تقاطع می گویند که متناظر با آمیزش موجودات زنده است.

تحقیقات بسیاری، تقاطع را به‌عنوان مکانیزم اصلی برای ایجاد تنوع در نظر گرفته‌اند. صرف نظر از درستی این دیدگاه، تقاطع مطمئناً یکی از ویژگی‌هایی است که الگوریتم ژنتیک و سایر الگوریتم‌های تکاملی را که از این عملگر استفاده می‌کنند از سایر الگوریتم‌های فراابتکاری، متمایز می‌کند.

تعریف ۲-۳-۹. نرخ تولید فرزندان جدید به روش تقاطع را نرخ تقاطع می‌گویند و آن را با P_c نشان می‌دهند که معمولاً در دامنه‌ی [۰.۵, ۱] قرار دارد.

اگر اندازه جمعیت PopSize باشد، آن‌گاه باید تعداد $P_c \times PopSize$ فرزند جدید را با استفاده از انجام عمل تقاطع روی افراد جامعه فعلی ایجاد کرد.

مثال ۲-۳-۱۰. اگر نرخ تقاطع را $P_c = ۰.۵۲$ فرض کنیم و ۴ کروموزوم داشته باشیم، در این‌صورت عملگر تقاطع بر روی دو کروموزوم اعمال می‌شود ($۰.۵۲ \times ۴ = ۲.۰۸ \simeq ۲$).

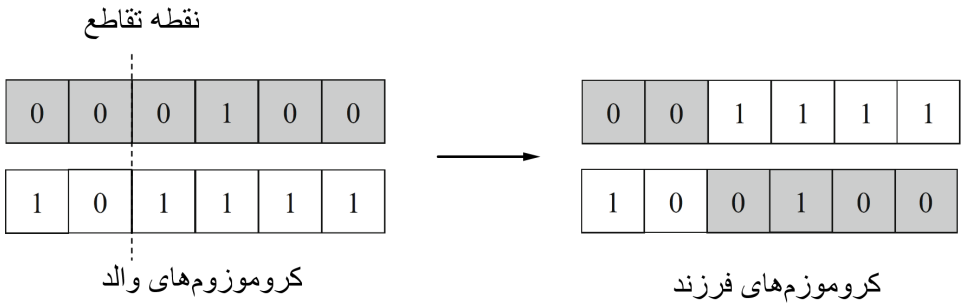
معمولاً دو والد انتخاب می‌شوند و سپس یک مقدار تصادفی از بازه‌ی [۰, ۱] تولید شده و با P_c مقایسه می‌شود. اگر مقدار کمتر بود، دو فرزند به‌وسیله‌ی تقاطع دو والد تولید می‌شوند؛ در غیر این‌صورت، فرزندان بدون آمیزش، یعنی با کپی کردن والدها، تولید می‌گردند. در نتیجه مجموعه بدست آمده از فرزندان، شامل برخی کپی‌ها از والدین و همچنین جواب‌های جدید است.

در الگوریتم ژنتیک، عمل تقاطع به شیوه‌های متفاوتی قابل انجام است. معمولاً سه روش استاندارد از عمل تقاطع برای نمایش‌های باینری کروموزوم‌ها استفاده می‌شود:

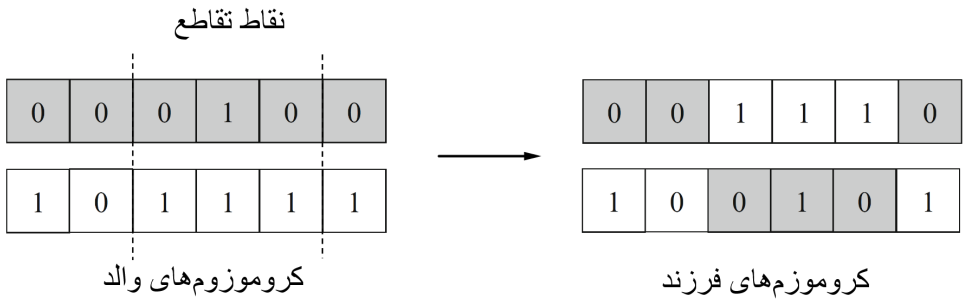
۲-۳-۳-۱-۱ تقاطع یک نقطه‌ای: فرض کنید L طول رشته بیتی و یا به‌عبارت دیگر بیانگر تعداد ژن‌ها در هر کروموزم باشد. تقاطع یک نقطه‌ای با انتخاب یک عدد تصادفی در بازه‌ی $[۰, L-۱]$ کار می‌کند و سپس هردوی والدین را در این نقطه می‌شکنند و دو فرزند را با جابجا کردن دنباله‌ی آن‌ها تولید می‌کند. شکل ۲-۹ نحوه تولید فرزندان با روش تقاطع تک نقطه‌ای را به تصویر می‌کشد.

۲-۳-۳-۱-۲ تقاطع n نقطه‌ای: تقاطع یک نقطه‌ای می‌تواند به آسانی به حالت کلی تقاطع n نقطه‌ای تبدیل شود. ابتدا n نقطه‌ی تقاطع تصادفی در بازه‌ی $[۰, L-۱]$ انتخاب می‌شوند. آن‌گاه ژن‌های مابین این نقاط از دو والد با یکدیگر جابجا شده و منجر به تولید فرزندان جدید می گردند که در شکل ۲-۱۰ برای $n = ۲$ این تقاطع نشان داده شده‌است.

۲-۳-۳-۱-۳ تقاطع یکنواخت: در مقایسه با دو عملگر قبلی، تقاطع یکنواخت با هر ژن به‌طور مستقل رفتار می‌کند و



شکل ۲-۹: تقاطع تک نقطه‌ای



شکل ۲-۱۰: تقاطع دو نقطه‌ای

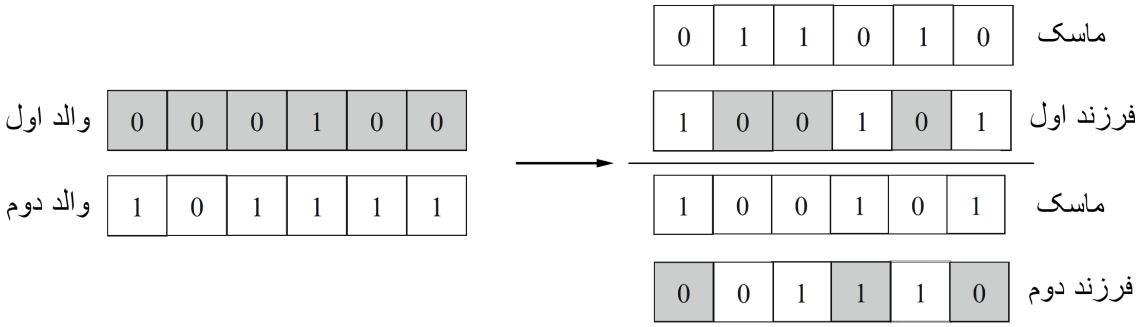
یک انتخاب تصادفی انجام می‌دهد که طبق آن مشخص می‌شود که هر ژن باید از کدام والد به ارث برده شود. در این روش، یک ماسک تقاطع هم طول با کروموزم‌های موجود باید ساخته شود.

تعریف ۲-۳-۱۱. به کروموزمی خالی که هر سلول آن به‌صورت تصادفی و یکنواخت با اعداد ۰ یا ۱ پر می‌شود، ماسک تقاطع می‌گویند.

هنگام پر کردن هر سلول از ماسک تقاطع، عدد ۱ با احتمال ۰.۵ و یا عدد ۰ با احتمال ۰.۵ قرار داده می‌شود. عدد ۱ به‌معنی انتخاب مقدار ژن متناظر با والد اول و عدد ۰ به‌معنی انتخاب مقدار ژن متناظر با والد دوم است. در شکل ۲-۱۱ نحوه تولید فرزندان به این روش نشان داده شده‌است.

۲-۳-۳-۲ عملگر جهش

عملگر جهش، بر روی یک کروموزم به‌کار می‌رود و یک فرزند جهش یافته را ایجاد می‌کند.



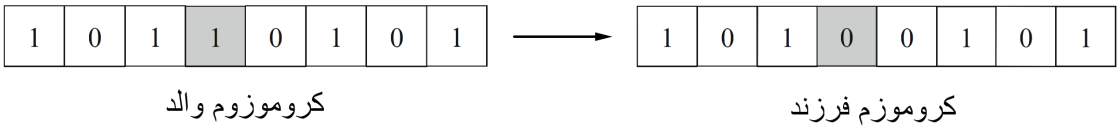
شکل ۲-۱۱: تقاطع یکنواخت

تعریف ۲-۳-۱۲. نرخ تولید فرزندان جدید به روش جهش را نرخ جهش می‌نامند و آن را با P_m نشان می‌دهند که مقدار آن بسته به نوع مسأله تعیین می‌گردد.

اگر اندازه جمعیت PopSize باشد، آن‌گاه باید تعداد $P_m \times PopSize$ فرزند جدید را با استفاده از انجام عمل جهش روی افراد جامعه فعلی ایجاد کرد.

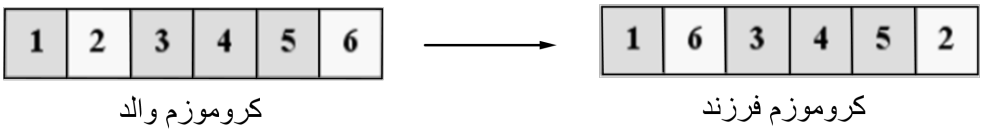
مثال ۲-۳-۱۳. اگر نرخ جهش $P_m = ۰.۰۱$ فرض شود، در این‌صورت روی جمعیتی با ۴ کروموزوم هیچ جهشی صورت نمی‌گیرد ($۰.۰۱ \times ۴ = ۰.۰۴ \simeq ۰$).

۲-۳-۳-۲-۱ جهش ژنی: رایج‌ترین عملگر جهش که برای نمایش‌های باینری کروموزوم‌ها مورد استفاده قرار می‌گیرد، معکوس کردن مقدار بیت است. این جهش، هر ژن را به‌طور مجزا مورد توجه قرار می‌دهد و به هر بیت اجازه می‌دهد تا از یک به صفر یا از صفر به یک تغییر کند. در شکل ۲-۱۲ نحوه تولید فرزند از روی ساختار والدش نشان داده شده‌است.



شکل ۲-۱۲: جهش ژنی

۲-۳-۳-۲-۱ جهش تعویض: این نوع جهش، با انتخاب تصادفی دو موقعیت ژن در رشته و جابجایی مقادیر آن‌ها با هم عمل می‌کند. در شکل ۲-۱۳ نحوه تولید فرزند جدید با استفاده از تعویض مقادیر ژن‌های دوم و ششم نشان داده شده‌است.



شکل ۲-۱۳: جهش تعویض

- به‌طور معمول، در الگوریتم ژنتیک رابطه $P_c + P_m = ۱$ برقرار است.

۲-۳-۴ جایگزینی

در الگوریتم ژنتیک، اندازه جمعیت ثابت است. بنابراین، باید مشخص شود که پس از تولید فرزندان از والدین انتخاب شده،کدام افراد از نسل فعلی و فرزندان جدید باید در نسل بعد حضور داشته باشند که با استفاده از مکانیزم انتخاب بازمانده صورت می‌گیرد. برخلاف انتخاب والد که نوعاً احتمالی است، انتخاب بازمانده معمولاً قطعی است. دو استراتژی برای انتخاب بازمانده عبارتند از:

۲-۳-۴-۱ جایگزینی مبتنی بر عمر: در این استراتژی، هر فرد برای تعداد تکرار معینی در جمعیت باقی می‌ماند و برازندگی افراد، در انتخاب آن‌ها برای جای‌گزینی، در نظر گرفته نمی‌شود.

۲-۳-۴-۱ جایگزینی مبتنی بر برازندگی: تمام استراتژی‌هایی که برای انتخاب والد در زیربخش ۲-۳-۲ تشریح شد، می‌تواند برای انتخاب افراد برای نسل بعدی هم به‌کار برده شوند. دو مکانیزم رایج برای این استراتژی وجود دارد:

- جایگزینی بدترین: λ تا از بدترین اعضای جمعیت، برای جایگزینی انتخاب می‌شوند که می‌تواند منجر به همگرایی زودرس شود که طی آن، جمعیت تمایل دارد خیلی سریع به سمت اعضای با برازندگی بالاتر همگرا شود. این نوع جای‌گزینی،در حالتی که جمعیت بسیار بزرگ است استفاده می‌شود.

- نخبه‌گرا: این رویکرد برای جلوگیری از حذف اعضای با برازندگی بالا استفاده می‌شود. بنابراین، اگر چنین فردی در گروهی که باید جایگزین شود، انتخاب شود و هیچ یک از فرزندان اضافه شده به جمعیت، دارای برازندگی بهتر یا برابر نباشند، آن‌گاه این فرد نگه‌داری می‌شود و یکی از فرزندان حذف می‌گردد.

۲-۳-۵ ساختار کلی GA

در الگوریتم ۲-۹ ساختار کلی الگوریتم ژنتیک آمده است. ضوابط توقف الگوریتم ژنتیک به شکل‌های مختلفی قابل طراحی است. یک آستانه‌ی از پیش تعیین شده برای مقدار تابع هدف و شرط رسیدن به این مقدار آستانه، می‌تواند به‌عنوان شرط خاتمه

الگوریتم ۲-۹ شبه کد الگوریتم ژنتیک

ورودی: تعداد جمعیت اولیه (n)، نرخ ترکیب (p_c)، نرخ جهش (p_m)، شرط خاتمه، روش‌های انتخاب.
خروجی: جواب بهینه.

۱: جمعیت n کروموزومی به طور تصادفی ایجاد کنید.
۲: برازندگی هر کروموزوم در جمعیت را ارزیابی کنید.
۳: مراحل زیر را تا زمانیکه شرط خاتمه برقرار شود انجام دهید:
۴: جمعیت جدیدی را تشکیل دهید:
۵: والدین را از میان جمعیت انتخاب کنید و با توجه به احتمال ترکیب، برای تشکیل فرزندان جدید آن‌ها را ترکیب کنید.
۶: با توجه به احتمال جهش فرزندان را مورد جهش قرار دهید.
۷: فرزندان جدید را در جمعیت بگنجانید.
۸: برازندگی هر کروموزوم را ارزیابی کنید.

در نظر گرفته شود. از آنجایی که الگوریتم‌های ژنتیک، احتمالی هستند و معمولاً هیچ تضمینی برای رسیدن به آستانه‌ی از پیش تعیین شده وجود ندارد، این شرط ممکن است هیچگاه تأمین نشود و الگوریتم هیچگاه متوقف نشود. بنابراین لازم است این شرط با شرط خاتمه‌ی دیگری همراه شود و به‌طور قطع متوقف می‌شود. برای این منظور گزینه‌های زیر مورداستفاده قرار می‌گیرند:

- گذشتن از ماکزیمم زمان CPU مجاز.
- رسیدن تعداد کل ارزیابی‌های برازندگی، به یک آستانه معین.
- برای یک دوره معین زمانی (یعنی برای یک تعداد تولید یا ارزیابی برازندگی) بهبودی حاصل نشود (یا بهبود کمی حاصل شود.).
- تنوع جمعیت، کمتر از یک آستانه معین قرار گیرد.
- تولید تعداد معینی از نسل‌ها.

به‌منظور دستیابی به درکی عمیق‌تر از مراحل اجرای الگوریتم ژنتیک، به مثال ۲-۳-۱۴ توجه نمایید [۲۲].

مثال ۲-۳-۱۴. مسأله کوله‌پشتی، مسأله‌ای است که یک کوهنورد در هنگام انتخاب وسایل لازم برای کوهنوردی با آن مواجه است. کوله‌پشتی این کوهنورد فضای محدودی دارد و کوهنورد مجبور است تعداد شئ محدودی را از بین n شئ که در اختیار دارد، انتخاب کند. به دلیل محدودیت فضا، امکان انتخاب همه اشیاء وجود ندارد. کوهنورد می‌خواهد اشیائی را انتخاب کند که بیشترین مطلوبیت را داشته باشد. این مسأله را می‌توان به صورت زیر فرمول‌بندی کرد:

$$\max Z = \sum_{i=1}^{n} x_i v_i \qquad (۱۰-۲)$$

$$s.t.$$

$$\sum_{i=۱}^{n} x_i w_i \leqslant W$$

$$x_i \in \{۰, ۱\} \quad i = ۱, ..., n$$

که در آن x_i یک متغیر صفر یا یک است که نشان‌دهنده حضور یا عدم حضور شئ iام در کوله است، v_i نشان‌دهنده ارزش (مطلوبیت) حضور شئ iام و w_i نشان‌دهنده حجم یا وزن شئ iام می‌باشد. محدودیت حداکثر حجم یا وزن کوله با W نمایش داده شده است. این مدل به دنبال حداکثر کردن ارزش کوله با رعایت محدودیت حجم یا وزن آن می‌باشد.

در این مثال فرض کنید ۱۰ شئ با مشخصات زیر در اختیار باشد و حداکثر وزن قابل قبول کوله برابر با ۲۰ کیلوگرم باشد.

جدول ۲-۵: مسأله کوله‌پشتی با ۱۰ شی

شماره شئ	۱	۲	۳	۴	۵	۶	۷	۸	۹	۱۰
مطلوبیت	۶۵	۸۵	۳۲	۹۸	۲۴	۲۶	۵۷	۴۳	۶۴	۲۱
وزن	۵	۵/۵	۲/۵	۶	۱/۵	۱/۸	۳/۵	۳	۴	۲

برای نشان دادن راه‌حل‌های این مسأله از کدگذاری صفر و یک استفاده می‌کنیم. برای مثال، از یک آرایه یک بعدی دارای ۱۰ ارزش به صورت $[x_۱, x_۲, ..., x_{۱۰}]$ استفاده می‌شود. در این آرایه مقادیر صفر یا یک قرار می‌گیرند که حضور یا عدم حضور شئ مرتبط در کوله را نشان می‌دهند. تعداد راه‌حل‌ها برابر $۲^{۱۰}$ خواهد بود که تعدادی از آن‌ها امکان‌پذیر نمی‌باشند.

با توجه به اینکه تعدادی از جواب‌های تولیدی امکان‌پذیر نمی‌باشند از روشی برای تبدیل جواب امکان‌ناپذیر به جواب امکان‌پذیر استفاده می‌شود. در این روش اشیاء داخل کوله بر اساس نسبت ارزش آن‌ها به وزن آن‌ها به صورت صعودی مرتب می‌گردند ($\frac{v_i}{w_i}$). سپس شروع به حذف اشیاء از جواب می‌کنیم تا یک جواب امکان‌پذیر بدست آید.

تولید جمعیت اولیه. در این مرحله، جمعیت اولیه با ۴ کروموزوم به صورت تصادفی تولید می‌گردد. جدول ۲-۶ جمعیت اولیه تولید شده را نشان می‌دهد.

جدول ۲-۶: جمعیت اولیه تولید شده

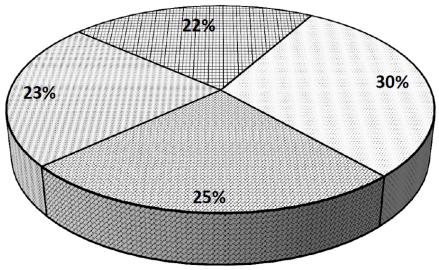
شماره کروموزوم	آرایه	وزن	امکان‌پذیری	$f(x)$	احتمال
۱	[۱۰۰۱۱۰۰۱۰۱]	۱۷/۵	بله	۲۵۱	۰/۲۵
۲	[۱۱۰۱۰۰۱۰۰۰]	۲۰	بله	۳۰۵	۰/۳۰
۳	[۰۰۱۰۱۱۱۰۱۱]	۱۵/۳	بله	۲۲۴	۰/۲۲
۴	[۰۱۰۰۰۱۱۰۱۰]	۱۴/۸	بله	۲۳۰	۰/۲۳
جمع				۱۰۱۰	۱

کروموزوم ۱ را در نظر بگیرید، مقدار ارزش این کد و وزن کوله به روش زیر محاسبه می‌گردد.

$$f([۱۰۰۱۱۰۰۱۰۱]) = ۶۵ + ۹۸ + ۲۴ + ۴۳ + ۲۱ = ۲۵۱$$

$$w([۱۰۰۱۱۰۰۱۰۱]) = ۵ + ۶ + ۱/۵ + ۳ + ۲ = ۱۷/۵$$

با توجه به اینکه در جمعیت اولیه وزن اشیاء کمتر از ۲۰ می‌باشد، جواب‌ها امکان‌پذیر هستند. احتمال انتخاب هر یک از کروموزوم‌ها به صورت حاصل تقسیم مقدار برازندگی آن کروموزوم بر مقدار برازندگی کل کروموزوم‌ها محاسبه شده است و به صورت درصد در شکل ۲-۱۴ نمایش داده شده است.



شکل ۲-۱۴: احتمال انتخاب کروموزم‌ها با استفاده از چرخ رولت

تکرار ۱: تولید جمعیت جدید. در این مرحله، بر اساس چرخ رولت کروموزوم‌های ۱ و ۲ و همچنین کروموزوم‌های ۲ و ۴ برای تولیدمثل انتخاب شده‌اند. از ترکیب تک نقطه‌ای استفاده می‌شود و نقطه تقاطع بر اساس یک قاعده تصادفی انتخاب شده است. دو کروموزوم اول در نقطه ۶ و دو کروموزوم دوم در نقطه ۴ ترکیب شده‌اند. با این روش ترکیب، ۴ کروموزوم فرزند تولید می‌شود.

جدول ۲-۷: فرزندان تولید شده در تکرار ۱

شماره والد	آرایه والد	شماره فرزند	آرایه فرزند	وزن	$f(x)$	توضیح
۱	[۱۰۰۱۱۰:۰۱۰۱]	۵	[۱۰۰۱۱۰۱۰۰۰]	۱۶	۲۴۴	
۲	[۱۱۰۱۰۰:۱۰۰۰]	۶	[۱۱۰۱۰۰۰۱۰۱]	۲۱/۵	—	اصلاح به کروموزوم ۹
۳	[۱۱۰۱:۰۰۱۰۰۰]	۷	[۱۱۰۱۰۱۱۰۱۰]	۲۵/۸	—	اصلاح به کروموزوم ۱۰
۴	[۰۱۰۰:۰۱۱۰۱۰]	۸	[۰۱۰۰۰۰۱۰۰۰]	۹	۱۴۲	
		۹	[۱۱۰۱۰۰۰۱۰۰]	۱۹/۵	۲۹۱	
		۱۰	[۰۱۰۱۰۰۱۰۱۰]	۱۹	۳۰۴	

با توجه به اینکه وزن کروموزوم فرزند شماره ۶ و ۷ بیشتر از ۲۰ است لذا این دو کروموزوم قابل قبول نمی‌باشند، این کروموزوم‌ها به شماره‌های ۹ و ۱۰ اصلاح می‌شوند. نحوه‌ی اصلاح کروموزوم شماره ۶ به صورت زیر است:

$$\frac{v_i}{w_i}(۱, ۲, ۴, ۸, ۱۰) = (\frac{۶۵}{۵}, \frac{۸۵}{۵/۵}, \frac{۹۸}{۶}, \frac{۴۳}{۳}, \frac{۲۱}{۲}) = (۱۳, ۱۵/۴۵, ۱۶/۳۳, ۱۴/۳۳, ۱۰/۵)$$

حذف اشیاء از کروموزوم ۶ به ترتیب ۱۰، ۱، ۸، ۲، ۴ می‌باشد، بر این اساس ابتدا شئ شماره ۱۰ حذف می‌شود و وزن کوله به ۱۹/۵ کیلوگرم کاهش می‌یابد. اکنون جواب شئ دیگری از کوله حذف نمی‌گردد و جواب اصلاح شده است.

جدول ۲-۸ فرزندان تولید شده را بعد از عمل جهش نشان می‌دهد. فرض کنید تنها فرزند اول و چهارم جهش پیدا کرده‌اند، در فرزند اول دو ژن و در فرزند چهارم یک ژن جهش یافته است. هر ژن به صورت تصادفی مستقل از سایر ژن‌ها شانس جهش خواهد داشت و این عمل بر اساس تولید عدد تصادفی انجام می‌گردد. برای هر ژن مستقل از سایر ژن‌ها، یک عدد تصادفی تولید و در صورتی که کمتر از عدد تصادفی جهش باشد، آن ژن جهش پیدا می‌کند.

جدول ۲-۸: فرزندان تولید شده در تکرار ۱ بعد از عمل جهش

شماره فرزند	آرایه فرزند	شماره ژن جهش یافته	شماره فرزند	آرایه فرزند	وزن	$f(x)$	توضیح
۵	[۱۰۰۱۱۰۱۰۰۰]	۲،۵	۱۱	[۱۱۰۱۰۰۱۰۰۰]	۲۰	۳۰۵	
۸	[۰۱۰۰۰۰۱۰۰۰]	-	۸	[۰۱۰۰۰۰۱۰۰۰]	۹	۱۴۲	
۹	[۱۱۰۱۰۰۰۱۰۰]	-	۹	[۱۱۰۱۰۰۰۱۰۰]	۱۹/۵	۲۹۱	
۱۰	[۰۱۰۱۰۰۱۰۱۰]	۳	۱۲	[۰۱۱۱۰۰۱۰۱۰]	۲۱/۵	—	اصلاح به کروموزوم ۱۳
			۱۳	[۰۱۰۱۰۰۱۰۱۰]	۱۹	۳۰۴	

از بین جمعیت فرزندان و جمعیت والدین، باید ۴ عضو انتخاب شود. فرض می‌شود که بهترین و بدترین کروموزوم به صورت ثابت به جمعیت بعدی منتقل شوند و ۲ کروموزوم دیگر به صورت تصادفی بر اساس مقدار برازندگی آن‌ها انتخاب گردند. جمعیت جدید در جدول ۲-۹ نمایش داده شده است. مراحل فوق تکرار می‌شود تا شرط توقف برآورده گردد.

جدول ۲-۹: جمعیت تولید شده بعد از تکرار ۱

شماره کروموزوم	آرایه	وزن	امکان‌پذیری	$f(x)$	احتمال
۲	[۱۱۰۱۰۰۱۰۰۰]	۲۰	بله	۳۰۵	۰/۲۹۳
۸	[۰۱۰۰۰۰۱۰۰۰]	۹	بله	۱۴۲	۰/۱۳۷
۱۳	[۰۱۱۱۰۰۱۰۱۰]	۱۹	بله	۳۰۴	۰/۲۹۱
۹	[۱۱۰۱۰۰۰۱۰۰]	۱۹/۵	بله	۲۹۱	۰/۲۷۹
جمع				۱۰۴۲	۱

۲-۳-۶ الگوریتم ژنتیک برای حل مسأله USAHLP

در مرجع [۲۹] الگوریتم ژنتیک برای مسأله USAHLP (این مسأله در بخش ۱-۵ مورد بررسی قرار گرفت) به صورت زیر توصیف شده است. همان‌طور که قبلاً بیان شد، الگوریتم ژنتیک یک الگوریتم جستجوگر برای یافتن جواب‌های نزدیک به جواب بهینه در فضاهای بزرگ است که از علم پیدایش جمعیت الهام گرفته شده‌است.

نمایش رشته‌ای

در رویکرد مبتنی بر الگوریتم ژنتیک ارائه شده، هر کروموزوم یا رشته بیتی (یعنی، یک نمونه جواب) متشکل از دو آرایه است که طول این آرایه‌ها برابر با تعداد گره‌های شبکه است. این آرایه‌ها عبارتند از:

- **آرایه هاب** [1]: شامل ۰ ها و ۱ هاست. مقدار ۱ نشان می‌دهد که گره موردنظر هاب است و مقدار ۰ نشان دهنده‌ی غیرهاب بودن آن گره است.
- **آرایه تخصیص** [2]: بیانگر تخصیص غیرهاب‌ها به هاب‌ها است. اگر گره i به گره k اختصاص داده شود، ورودی آن در آرایه تخصیص مقدار k می‌گیرد. علاوه بر این، هر هاب در این آرایه به خودش اختصاص می‌یابد.

تولید جمعیت اولیه

در مرحله تولید جمعیت اولیه، تعداد از پیش تعریف شده‌ای از رشته‌ها ایجاد می‌شوند. اولین فاز مربوط به انتخاب هاب‌هاست. سپس، هر گره غیرهاب بر اساس مقادیر فاصله‌ای به نزدیک‌ترین هاب اختصاص داده می‌شود. همان‌طور که قبلاً نیز بیان شد، تعداد هاب‌ها در مسئله‌ی USAHLP یک متغیر تصمیم است. برای ٪۷۵ جمعیت اولیه، تعداد هاب‌ها به‌صورت تصادفی از محدوده‌ی $[۱, \ldots, n/۴]$ تعیین می‌شود، که در آن n تعداد کل گره‌ها است. برای ٪۲۵ باقی‌مانده از جمعیت اولیه، تعداد هاب‌ها از محدوده‌ی $(n/۴, \ldots, n/۲]$ مشخص خواهند شد. با این استراتژی، تعداد کل هاب‌های شبکه بیش از نصف اندازه‌ی مسأله مورد نظر خواهدبود. پس از تعیین تعداد هاب‌ها، این هاب‌ها در آرایه هاب قرار می‌گیرند.

با انتخاب هاب‌ها، مقدار کل جریان در هر گره i ، که برابر با $O_i + D_i$ است، محاسبه می‌شود. همان‌طور که می‌دانیم، O_i و D_i به ترتیب بیانگر مقدار کل ترافیک خروجی و ورودی گره i هستند. سپس، گره‌ها در لیستی به نام لیست جریان [3] در جهت کاهش کل جریان‌ها مرتب می‌شوند. برای ٪۷۵ از جمعیت اولیه، هاب‌ها در آرایه هاب از دو سوم گره‌های لیست جریان با شروع از گره با بالاترین جریان کل انتخاب می‌گردند. این استراتژی با احتمال زیادی زمینه را برای تبدیل گره‌ای با بیشترین مقدار جریان به یک گره هاب فراهم می‌کند. برای ٪۲۵ باقی مانده از جمعیت، هاب‌ها از کل مجموعه گره‌ها انتخاب می‌شوند. این مقادیر درصدتوسط مجموعه‌ای از آزمایشات برای تعیین پارامترهای کنترل GA تعیین شده‌اند.

مثال ۲-۳-۱۵. اگر اندازه جمعیت برابر با ۲۰ و تعداد گره‌ها برابر با ۲۴ باشد، ۱۵ نفر دارای شش هاب هستند که از بین آن گره‌هایی که بیشترین جریان را دارند، انتخاب شده‌اند؛ و پنج نفر باقی مانده دارای ۱۲ هاب خواهند بود که از همه‌ی گره‌های سیستم برگزیده خواهند شد.

در آزمایش‌های انجام گرفته، تعداد ۲۰۰ نسل لحاظ شده‌است. علاوه بر این، اگر بهترین مقدار برازندگی، بیش از یک مقدار تحمل از پیش تعریف شده، یعنی ۰.۰۰۰۱ در آزمایشات انجام گرفته، در طول ۵۰ نسل گذشته، تغییر نکند، یکی دیگر از معیارهای توقف در نظر گرفته خواهد شد.

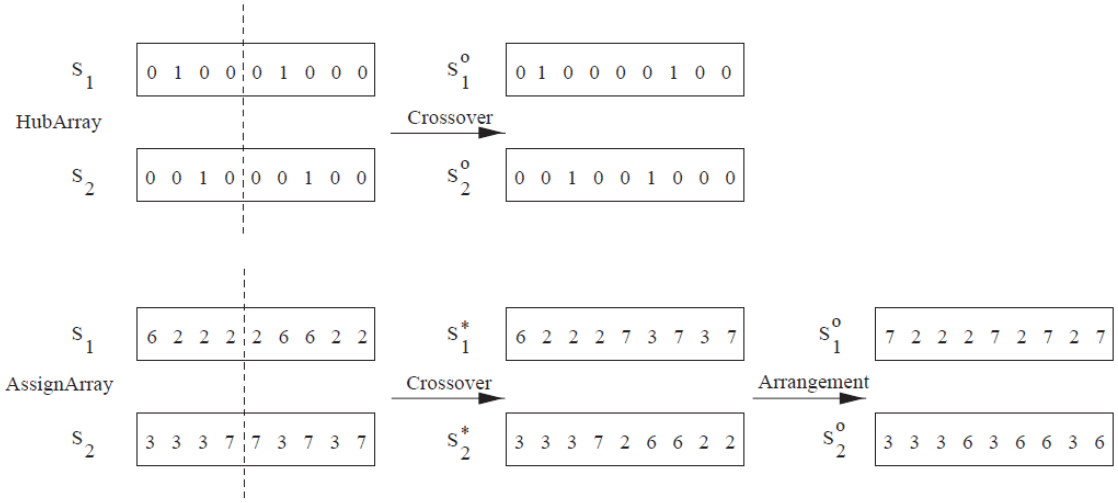
عملگر تقاطع

انتخاب مبتنی بر تناسب برازندگی با نمونه‌گیری چرخ رولت، در چارچوب الگوریتم ژنتیک موردنظر در نظر گرفته شده‌است. مقدار برازندگی یک فرد، با اثر متقابل کل هزینه فرد در معادله ؟؟ تعیین می‌شود. در آزمایشات، استراتژی نخبه‌گرایی با قرار دادن بهترین کروموزوم در هر نسل به جای بدترین فرد با توجه به هزینه کل اعمال می‌گردد. با فرض نقطه تقاطع یکسان که به‌طور تصادفی انتخاب شده‌است، از عملگر تقاطع تک-نقطه‌ای روی آرایه هاب و آرایه تخصیص رشته‌های ورودی استفاده

[1] HubArray [2] AssignArray [3] flow-list

می‌شود. تولید فرزندان با ترکیب بخش‌های چپ و راست صورت می‌گیرد، که در صورت لزوم یک مرحله برای تنظیم فرزندان دنبال خواهد شد. پس از اعمال عملگر تقاطع بر روی آرایه‌های هاب، اگر هر کدام از فرزندان هیچ هابی نداشته باشد، یا اگر تعداد هاب‌ها برابر تعداد گره‌ها باشد، آنگاه هر دو فرزند دور انداخته خواهندشد. برای هر یک از فرزندان آرایه‌های تخصیص، اگر گره i به گره دیگری که به دلیل انجام عمل تقاطع روی آرایه‌های هاب، دیگر گره‌ای هاب نیست تخصیص داده شود، آنگاه گره i با توجه به مقادیر فاصله‌ای به نزدیکترین هاب تخصیص مجدد می‌یابد.

مثال ۲-۳-۱۶. شکل ۲-۱۵، استفاده از عملگر تقاطع را در قالب یک مثال نشان می‌دهد. در ابتدا گره‌های ۲ و ۶ برای گره غیرهاب S_1 و گره‌های ۳ و ۷ برای گره غیرهاب S_2 به‌عنوان گره‌های هاب در نظر گرفته شده‌اند. بعد از اعمال عملگر تقاطع روی آرایه‌های هاب، اولین فرزند یعنی S_1^o گره‌های هاب ۲ و ۷ و دومین فرزند یعنی S_2^o گره‌های هاب ۳ و ۶ را اتخاذ می‌کند. همچنین، بعد از بکارگیری این عملگر روی آرایه‌های تخصیص، از آنجایی که گره ۳ و گره ۶ در آرایه هاب S_1^o هاب نیستند، تخصیص گره‌های ۱، ۶ و ۸ در S_1^* نیاز به تنظیم مجدد براساس مقادیر فاصله‌ای دارد. تنظیم مشابهی برای آرایه تخصیص S_2^* انجام گرفته است.



شکل ۲-۱۵: بکارگیری عملگر تقاطع

عملگر جهش

عملگر جهش فقط برای بخش تخصیص گره‌ها اعمال می‌شود. دو عملگر جهش بکار برده شده‌اند:

- **تغییر** [۱]: این عملگر یک گره غیرهاب را انتخاب کرده و آن‌را مجدداً به هاب دیگری که به‌طور تصادفی انتخاب شده است، اختصاص می‌دهد. اگر تنها یک هاب در رشته وجود داشته باشد، این عملگر جهش اعمال نخواهد شد.
- **مبادله** [۲]: این عملگر دو گره غیرهاب را به‌طور تصادفی انتخاب کرده و تخصیص‌شان را معاوضه می‌کند. از آن‌جایی که حداقل دو جفت از هاب‌ها و غیرهاب‌ها موردنیاز است، اگر تنها یک هاب یا تنها یک غیرهاب وجود داشته باشد، آنگاه حرکت مبادله قابل اجرا نیست.

[۱] shift [۲] exchange

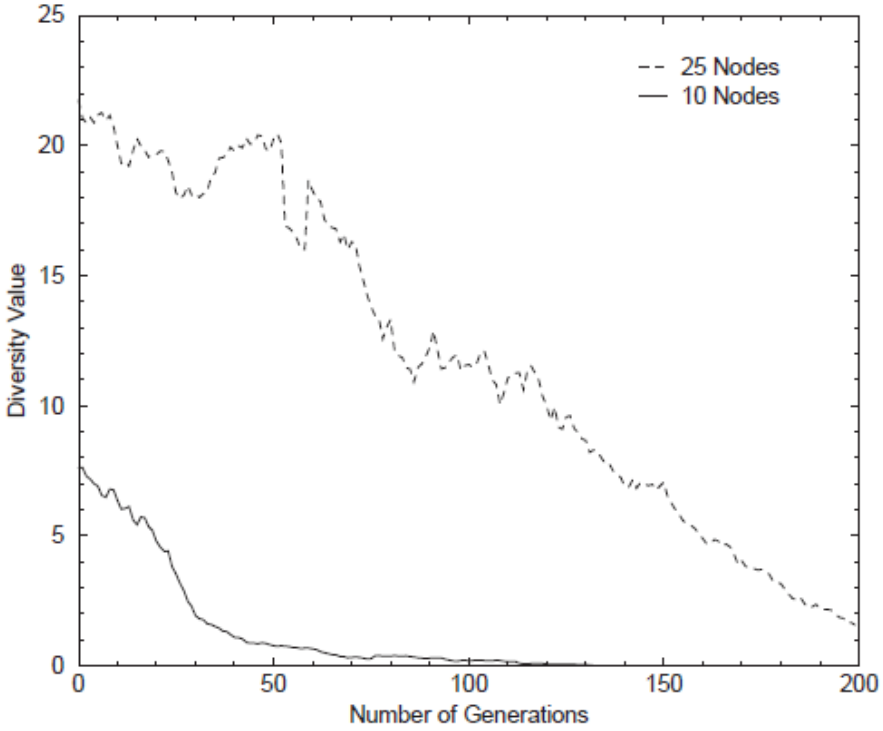
این دو روش برای فرد انتخاب شده بکار می‌روند و نتیجه با مقدار برازندگی بهتر با یک نرخ جهش از پیش تعریف شده انتخاب می‌شود.

سیرتکاملی تنوع

تنوع جمعیت دارای اهمیت زیادی در عملکرد الگوریتم‌های ژنتیکی است و کاهش سریع تنوع در جمعیت‌ها به همگرایی زودرس کمک می‌کند. به‌منظور مطالعه سیر تکاملی تنوع در طول اجرای یک الگوریتم ژنتیک، باید اندازه‌ای برای تنوع ارائه شود. در اینجا فاصله‌ی همینگ [1] روی آرایه‌های تخصیص برای اندازه‌گیری تنوع در نظر گرفته شده‌است.

تعریف ۲-۳-۱۷. فاصله‌ی همینگ برای دو رشته با طول مساوی، برابر تعداد مکان‌هایی است که سمبول‌های متناظر متفاوت هستند. به عبارت دیگر، کمترین تعداد جایگزینی‌هایی است که یک رشته به یک رشته دیگر تغییر پیدا کند، یا تعداد خطاهایی که یک رشته به رشته دیگر تبدیل گردد.

اگر دو آرایه‌ی تخصیص S_1 و S_2 داده شده باشند، آنگاه فاصله‌ی همینگ بین آن‌ها برابر با تعداد تغییرات در انتساب‌های موردنیاز برای تبدیل S_1 به S_2 است. تنوع یک جمعیت داده شده، با فاصله متوسط بین فرد با بهترین جواب و تمام افراد دیگر در جمعیت تعریف می‌شود. این روش همچنین تنوع بین دو جواب را اندازه گیری می‌کند، حتی اگر از مجموعه‌ی یکسانی از گره‌ها به عنوان هاب استفاده کرده باشند. در شکل ۲-۱۶، تنوع جمعیت برای دو مسئله با اندازه‌های مختلف از مجموعه داده



شکل ۲-۱۶: نمودار نمایش مقدار تنوع جمعیت با تغییر شماره نسل‌ها

CAB فراهم شده‌است. جمعیت اولیه در مسائل با اندازه ۱۰ و ۲۵ به ترتیب دارای مقدار تنوع ۸ و ۲۲ است. این مقادیر تنوع

[1] hamming distance

نشان می دهند که فاز جمعیت اولیه جواب‌های کاملا متفاوتی را برای جلوگیری از همگرایی زودرس ایجاد می‌کند. مقادیر تنوع برای دو آزمون داده شده با افزایش تعداد نسل‌ها کاهش می‌یابد.

فصل ۳

طرح یک الگوریتم ممتیک کارامد

الگوریتم‌های ممتیک، تکنیک‌های بهینه‌سازی هستند که بر مبنای ترکیب سینرژیکی ایده‌های گرفته شده از راه‌حل‌های الگوریتمی متفاوت مانند جستجوی مبتنی بر جمعیت (تکنیک‌های تکاملی) و جستجوی محلی عمل می‌کنند.

- زمانی که دو یا چند عنصر با هم برهمکنش (تعامل) داشته باشند اثری بوجود می‌آید که اگر این اثر از مجموع اثرهایی که هر کدام از آن عناصر جداگانه می‌توانستند بوجود آورند بیشتر شود، پدیده سینرژیک [۱] رخ داده‌است.

مفهوم "مم" [۲] از فلسفه اقتباس شده‌است و به‌عنوان واحد انتقال فرهنگی در نظر گرفته می‌شود. به‌عبارت دیگر، ایده‌های پیچیده می‌توانند به "مم"ها یا الگوهای رفتاری که درون یک جامعه پخش شده و تغییر می‌کنند، تجزیه شوند. فرهنگ، از این طریق، به‌طور مداوم تکامل می‌یابد و تمایل به پیشرفت‌های چشمگیر دارد. ایده‌های قوی متمایلند که مقاومت کنند و در جامعه تبلیغ شوند در حالی‌که ایده‌های ضعیف انتخاب نمی‌شوند و تمایل به ناپدید شدن دارند. استعاراً، ایده‌ها عملگرهای جستجو هستند به‌طوریکه مناسب‌ترین آن‌ها به‌کار گرفته می‌شوند و ایده‌های نامساعد احتمالاً ناپدید خواهند شد.

به منظور دستیابی به درکی عمیق از نقش و نیاز به الگوریتم‌های ممتیک ، ضروری است تا زمینه‌ی تاریخی این الگوریتم‌ها مورد بررسی قرار گیرد. در سال ۱۹۸۸ یعنی زمانی‌که اولین الگوریتم‌های ممتیک معرفی شدند، الگوریتم‌های ژنتیک در میان دانشمندان علوم کامپیوتر بسیار محبوب بودند و تحقیقات مربوط به آن‌ها متمایل به طراحی الگوریتم‌های دارای عملکرد برتر نسبت به تمام الگوریتم‌های دیگر موجود در ادبیات بود که این رویکرد در بسیاری از متون مشهور منتشر شده در آن سال‌ها قابل مشاهده است. الگوریتم ممتیک برخلاف تمامی الگوریتم‌هایی که در آن زمان پیشنهاد شدند، یک الگوریتم خاص نبود، اما بسیار کلی‌تر از یک الگوریتم بهینه‌سازی عمل می‌کرد. از آنجایی که الگوریتم‌های ممتیک از ترکیب الگوریتم‌های جستجوی سراسری و محلی تشکیل شده‌اند، نماینده یک کلاس گسترده و قابل انعطاف از الگوریتم‌هایی هستند که به نوعی شامل کارهای قبلی انجام گرفته در الگوریتم‌های تکاملی بوده و بنابراین یک فلسفه جدید در بهینه سازی را تشکیل داده‌اند.

احتمالاً به‌دلیل محتویات بیش از حد نوآورانه بوده‌است که این الگوریتم‌ها مجبور شدند حدود یک دهه با شک و تردید جامعه علمی که مکرراً رویکرد ممتیک را به‌عنوان یک امکان بالقوه در بهینه‌سازی رد می‌کردند، مواجه شوند. از سال ۱۹۹۷

---

[۱] Synergy [۲] meme

تا به الان، محققان بهینه‌سازی به‌طور چشمگیری دیدگاه خود را در مورد این موضوع تغییر داده‌اند. با توجه به افزایش علاقه به هدف کلی الگوریتم‌های بهینه‌سازی، الگوریتم ممتیک در پایان دهه ۹۰ به یک الگوریتم مهم تبدیل شد.

۳-۱ حل مسئله مکانیابی با الگوریتم ممتیک

در بسیاری از مسائل بهینه‌سازی ترکیبیاتی به منظور بازده مؤثر ابزارهای بهینه‌سازی، استفاده از برخی استراتژی‌های بهبوددهنده داخل یک روش ابتکاری، ضروری به نظر می‌رسد. الگوریتم‌های ممتیک روش‌هایی فراابتکاری‌اند که تلفیقی از جستجوی محلی ابتکاری و مفهوم تکاملی سنتی را بیان می‌کنند. بعبارتی، این الگوریتم‌ها یک تکنیک جستجوی محلی را برای هدایت الگوریتم تکاملی به سمت مناطق امیدبخش فضای جستجو استفاده می‌نمایند.

در مرجع [۲۱] الگوریتم ممتیک بسیار کارآمدی برای حل مسئله مکانیابی USAHLP مطرح شده‌است. انگیزه اصلی پژوهش ارائه شده در این مطالعه،از پتانسیل استراتژی تکاملی بیان شده در بخش ۲-۳-۶ گرفته شده‌است. در بخش تکاملی الگوریتم، از عملگرها و پارامترهای مناسب و استراتژی‌های تکاملی سازگار با مسئله استفاده می‌شود که از طریق یک سری آزمایش‌های محاسباتی انتخاب شده‌اند تا تعادل بین کارایی الگوریتم و کیفیت جواب‌ها به‌دست آید. این عملگرها عبارتند از:

- انتخاب مسابقه‌ای
- عملگر تقاطع دو نقطه‌ای: به‌طور تصادفی دو نقطه تقاطع در کدهای ژنتیک والدین را انتخاب کرده و مواد ژنتیکی بین آن‌ها را مبادله می‌کند و مولد دو فرزند است. در پیاده‌سازی الگوریتم ممتیک، ٪۸۵ افراد انتخاب شده مشمول عملگر تقاطع هستند؛
- عملگر جهش با بیت‌های یخ زده: فرض کنید بیت‌ها با احتمال $P_m = ۰/۴/m$ تغییر داده می‌شوند که m طول کد ژنتیک است. در طول اجرای الگوریتم ممتیک ممکن است تقریباً همه افراد جمعیت در یک موقعیت خاص دارای مقدار بیت یکسانی باشند. این بیت‌ها یخ‌زده نامیده می‌شوند. اگر تعداد بیت‌های یخ‌زده l باشد، فضای جستجو $۲l$ بار کوچکتر شده و امکان همگرایی زودرس به سرعت افزایش می‌یابد. عملگر انتخاب و عملگر تقاطع نمی‌توانند مقدار بیتی یک بیت یخ زده را تغییر دهند و اغلب نرخ جهش پایه به اندازه کافی بزرگ نیست تا قسمت‌های از دست رفته فضای جستجو را بازیابی کند. از طرفی، اگر نرخ جهش پایه به‌طور قابل توجهی افزایش یابد، الگوریتم ژنتیک به یک جستجوی تصادفی تبدیل می‌شود. به همین دلیل، نرخ جهش تنها در بیت‌های یخ‌زده بیش از چند بار افزایش نمی‌یابد. بنابراین، برای بیت‌های یخ‌زده نرخ جهش در فاکتور ۳.۵ ضرب خواهد شد.

علاوه بر این، از آنجایی که به نظر می‌رسد یک الگوریتم ژنتیک محض برای نمونه‌های بزرگ مسئله کارساز نیست، دو جستجوی محلی ابتکاری ایجاد شده و در چارچوب الگوریتم تکاملی به‌کار می‌روند. با این حال، امکان همگرایی نابهنگام

به یک بهینه محلی وجود دارد. به منظور حل این مشکل، چندین استراتژی جهت افزایش تنوع افراد به‌کار برده می‌شوند و الگوریتم را از افتادن در دام بهینه محلی حفظ می‌کنند. استراتژی‌های مختلفی که در پیاده‌سازی الگوریتم ممتیک مورد استفاده قرار می‌گیرند تا بتوانند الگوریتم را با موفقیت در فضای جستجو هدایت کنند و باعث بهبود بهره‌وری آن شوند عبارتند از:

- از رویکرد حالت پایدار با افراد نخبه به‌عنوان استراتژی جایگزینی استفاده خواهد شد. در طول هر نسل، فقط بدترین ۱/۳ جمعیت جایگزین می شود، در حالیکه ۲/۳ جمعیت باقی مانده، افراد نخبه‌ای هستند که برای نسل بعدی زنده مانده و ژن‌های بسیار مناسب را حفظ می‌کنند؛
- عملگر انتخاب، از ورود افراد تکراری به نسل بعد جلوگیری می‌کند. این استراتژی به حفظ تنوع مواد ژنتیکی کمک کرده و الگوریتم را به دور از دام بهینه محلی نگه می‌دارد؛
- افرادی که مقدار هدف یکسان اما کدهای ژنتیکی متفاوتی دارند، ممکن است پس از تعداد معینی نسل بر جمعیت مسلط شوند. اگر کدهای آن‌ها نیز مشابه باشد، باعث همگرایی زودرس الگوریتم ممتیک خواهند شد. به همین دلیل، بازگشت چنین افرادی به چمعیت محدود می‌شود؛
- از دو معیار توقف استفاده می‌شود:
  - G_{max}: حداکثر تعداد نسل
  - R_{max}: حداکثر تعداد تکرار بهترین فرد

در این بخش با طراحی دو الگوریتم جستجوی محلی ابتکاری کارآمد در چارچوب یک الگوریتم تکاملی، به بهبود دو بخش مکانیابی و تخصیص مسئله USAHLP پرداخته می‌شود. به‌طورخلاصه بعد از این‌که جمعیت اولیه تولید شد، در هر نسل الگوریتم ممتیک به محاسبه تابع هدف تک تک افرد اقدام می‌گردد. سپس جستجوهای محلی HeurLoc و HeurAlloc روی دو زیرمجموعه انتخاب شده از جمعیت اجرا می‌شوند که این زیرمجموعه‌ها در حالت کلی متفاوت‌اند اما ممکن است برابر یا منطبق بر جمعیت نیز باشند. جستجوی محلی HeurLoc بدون تغییردادن تعداد هاب‌های تأسیس شده، مبادرت به یافتن مکان‌های هاب بهتر در هر فرد اولین زیرمجموعه می‌کند. همچنین، جستجوی محلی HeurAlloc که روی یک زیرمجموعه ثابت از هاب‌ها به‌کار انداخته می‌شود سعی در بهبود تخصیص گره‌های غیرهاب به گره‌های هاب برای هر فرد در دومین زیرمجموعه دارد. بعد از بخش بهسازی جستجوی محلی، عملگر انتخاب به منظور انتخاب جفت جواب‌های والدینی برای انجام عملگر تقاطع استفاده می‌شود. جواب‌های کاندید جدید با ترکیب مجدد این والدین ایجاد شده و با یک احتمال مشخص تغییر داده می‌شوند. تکرار این روند تا زمان برقراری ضابطه توقف ادامه می‌یابد. چارچوب کلی الگوریتم ممتیک برای حل مسئله USAHLP به قرار زیر است:

- تولید جمعیت اولیه؛

- **تکرار مراحل زیر تا زمان برقراری ضابطه توقف**
- محاسبه تابع هدف برای هر فرد در جمعیت؛
- اجرای جستجوی محلی HeurLoc ؛
- اجرای جستجوی محلی HeurAlloc ؛
- انتخاب؛
- تقاطع؛
- جهش؛

در ادامه تمام جنبه‌های الگوریتم ممتیک پیشنهاد شده با جزئیات شرح داده خواهد شد.

۳-۱-۱ مقداردهی اولیه الگوریتم ممتیک

فرض کنید شماره‌گذاری گره‌های شبکه داده شده به‌صورت $I = ۰, ۱, \cdots, n-۱$ باشد. با توجه به این واقعیت که بسیاری از مسائل به‌طور طبیعی با متغیرهای باینری فرمولبندی می‌شوند، در بهینه‌سازی گسسته نیز نمایش باینری بسیار محبوب است. بنابراین، از یک بردار باینری با n ژن که هر ژن آن به یک گره متفاوت در شبکه مربوط است برای نمایش یک جواب کاندید استفاده می‌شود. همه‌ی ژن‌ها، که به صورت $۰, ۱, \cdots, n-۱$ به‌عنوان گره‌های شبکه نشان داده می‌شوند، طول یکسانی دارند و شامل $nbit$ ذره هستند. هر ژن به دو بخش تقسیم می‌شود که اولین بخش آن شامل ۱ ذره و دومین بخش آن شامل $nbit - ۱$ ذره است. به‌عنوان مثال، اگر اولین ذره k امین ژن مقدار ۱ بگیرد به این معنی است که یک هاب در k امین گره شبکه واقع شده‌است ($Z_{kk} = ۱$). این در حالیست که جاگیری ۰ در همان موقعیت ذره نشان دهنده‌ی غیرهاب بودن k امین گره خواهد بود ($Z_{kk} = ۰$). دومین بخش هر ژن، بیان کننده کدگذاری باینری عددصحیح از مجموعه $\{۰, ۱, \cdots, n-۱\}$ است که مشخص می‌کند کدام گره هاب به گره غیرهاب موردنظر تخصیص داده شده‌است. طبیعتاً، از آن‌جایی که هر هاب به خودش تخصیص می‌یابد می‌توان دومین بخش آن ژن را نادیده گرفت. برای کدگذاری یک عدد صحیح کمتر از n در دومین بخش هر ژن، به تعداد $\log_۲ n$ ذره نیاز داریم. لذا، طول کد ژنتیک هر فرد به‌صورت تابعی از n تعریف می‌شود، یعنی، $length = n.nbit$ که اندازه هر ژن به تنهایی برابر $nbit = \log_۲ n + ۱$ است. باید توجه داشته باشید در رویه کدگذاری باینری به‌کار رفته کلیه اطلاعات درباره یک گره در ژن مربوط به آن در کدژنتیک طبقه‌بندی می‌شود که برای اجرای مناسب جنبه‌های دیگر الگوریتم ممتیک حائز اهمیت است. تولید اولین ذره هر ژن که بیانگر هاب یا غیرهاب بودن گره موردنظر است به‌طور تصادفی صورت می‌گیرد که این رویکرد باعث ایجاد حداکثر تنوع ژنتیکی شده و شیب بهتر تابع هدف را فراهم می‌کند.

با توجه به عدم وجود محدودیت ظرفیت در مدل مورد بحث، معقولانه به نظر می‌رسد که انتظار داشته باشیم گره‌های غیرهاب به نزدیک‌ترین هاب‌ها و به ندرت به دورترین آن‌ها اختصاص یابند. از این‌رو، مطلوب است که جستجو را به نزدیک‌ترین هاب‌ها از نظر هزینه حمل‌ونقل هدایت کنیم و کمترین احتمال را برای اختصاص هر گره غیرهاب به دورترین هاب‌ها در نظر بگیریم و همچنین، بهتر است که دومین بخش هر ژن که حامل اطلاعات تخصیص گره غیرهاب جاریست شامل تعداد زیادی صفر باشد. به همین دلیل، اولین ذره در دومین بخش ژن با احتمال $۰.۵/n$ برابر ۱ خواهد شد و ذره‌هایی که به دنبال آن می‌آیند با نصف احتمال قبلی‌شان یعنی $۰.۲۵/n$ و $۰.۱۲۵/n$ و ... مقدار ۱ خواهند گرفت. همان‌طور که قبلاً نیز اشاره شد، با توجه به مقدار اولین ذره هر ژن در کد ژنتیکی فرد می‌توان مجموعه‌ی هاب‌های تعیین مکان شده را مشخص کرد. بعبارتی،

$$k \in \{۰, ۱, \cdots, n-۱\} \qquad s.t \quad Z_{kk} = ۱$$

پس از تعیین اندیس هاب‌ها از کد ژنتیکی، برای هر گره غیرهاب با توجه به هزینه حمل‌ونقل بین آن گره و گره‌های هاب مبادرت به تنظیم آرایه‌ای صعودی از این اندیس‌ها می‌نماییم. طول این آرایه برابر n_{hub} است و شمارنده آن از صفر شروع می‌شود. اندیسی که به گره غیرهاب تخصیص داده شده (متغیر $Z_{ik}; i \neq k$) از بخش دوم ژن استخراج می‌شود. اگر عدد صحیحی که در بخش دوم ژن در سیستمی باینری نمایش داده شده‌است به پیمانه n_{hub} دارای ارزش l باشد به این معناست که l امین عنصر آرایه مرتب شده اندیس هاب موردنظر است.

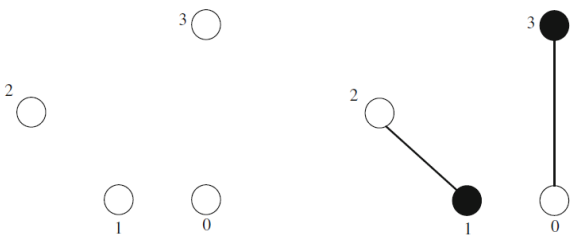
هر فردی که به این روش تولید شود و حداقل دارای یک گره هاب باشد، به‌عنوان جوابی شدنی به حساب می‌آید. بعلاوه، احتمال تولید فردی که هیچ گره هابی نداشته باشد بسیار کم است و اساساً چنین افردی دور انداخته می‌شوند. به دو دلیل، امکان حضور افرادی با کدگذاری‌های باینری متفاوت اما راه‌حل مشابه (مکانیابی و تخصیص) در جمعیت وجود دارد:

۱. دومین بخش ژن گره‌های هاب، همیشه نادیده گرفته می‌شود؛

۲. تخصیص گره غیرهاب به گره هاب از عدد صحیح استخراج شده از دومین بخش ژن به پیمانه n_{hub} محاسبه می‌شود؛

با توجه به دومین دلیل، اگر بین دو نسل با تغییر شماره n_{hub} برای فردی مواجه شویم تخصیص آن نیز تغییر خواهد کرد حتی اگر بخش دوم هر یک از ژن‌هایش تغییر نکند.

مثال ۳-۱-۱. در شکل زیر شبکه‌ای ساده با $n = ۴$ گره نشان داده شده‌است که رنگ سیاه معرف گره هاب است.



شکل ۳-۱: شبکه ساده با ۴ گره (.)

گره‌ها به صورت $(۰, ۱, ۲, ۳)$ مشخص شده‌است و ماتریس جریان W و ماتریس هزینه حمل و نقل C نیز داده شده‌اند:

$$\mathbf{W} = \begin{pmatrix} ۱ & ۲ & ۳ & ۴ \\ ۶ & ۲ & ۱ & ۳ \\ ۴ & ۳ & ۲ & ۲ \\ ۱ & ۱۰ & ۵ & ۱ \end{pmatrix} \quad , \quad \mathbf{C} = \begin{pmatrix} ۰ & ۱ & ۴ & ۳ \\ ۱ & ۰ & ۲ & ۴ \\ ۴ & ۳ & ۰ & ۴ \\ ۳ & ۴ & ۴ & ۰ \end{pmatrix}$$

هزینه‌های ثابت تأسیس هاب‌ها در هر گره با بردار $f = (۱۰, ۲۰, ۳۰, ۴۰)$ بیان می‌شوند. همچنین، χ = ۱ و α = ۰.۵ و δ = ۱.۵ پارامترهای هزینه‌های حمل و نقل هستند. کد ژنتیکی فرد به‌صورت $\overrightarrow{|\ ۱\ ۰\ ۱\ |\ ۰\ ۰\ ۰\ |\ ۱\ ۱\ ۰\ |\ ۱\ ۰\ ۱\ |}$ است. همان‌طور که مشاهده می‌کنید این کد متشکل از چهار ژن به طول $nbit = \lfloor \log_۲ ۴ \rfloor + ۱ = ۳$ است که در مجموع کدی به طول ۱۲ بیت ارائه می‌دهد. با توجه به اولین بیت هر ژن، متوجه انتخاب گره‌های ۱و ۳ به‌عنوان گره‌های هاب می‌شویم؛ بعبارتی، $Z_{۱۱} = Z_{۳۳} = ۱$ و $n_{hub} = ۲$. در ادامه، براساس هزینه‌های حمل و نقل بین گره‌های هاب تعیین مکان شده و گره‌های غیرهاب ۰ و ۲ $(Z_{۰۰} = Z_{۲۲} = ۰)$ آرایه‌ای صعودی از هاب‌ها مرتب می‌شود. با در نظر گرفتن ماتریس هزینه C نمای این آرایه برای گره غیر هاب ۰ به‌صورت $\begin{pmatrix} ۱ \\ ۳ \end{pmatrix}$ و برای گره غیرهاب ۲ نیز به‌صورت $\begin{pmatrix} ۱ \\ ۳ \end{pmatrix}$ خواهد بود.

با دقت در بخش دوم اولین ژن که مربوط به گره غیرهاب ۰ است، نمایشی از سیستم باینری عدد صحیح ۳ به چشم می‌خورد. بعبارت دیگر، عدد صحیح ۳ از بخش دوم ژن استخراج می‌گردد. در نتیجه داریم

$$۳ \overset{n_{hub}}{\equiv} ? \qquad \Longrightarrow \qquad ۳ \overset{۲}{\equiv} ۱$$

از آنجایی که شماره‌گذاری آرایه مرتب شده از صفر شروع می‌شود، گره غیرهاب ۰ به دومین عنصر این آرایه یعنی گره هاب ۳ تخصیص می‌یابد. بعبارتی، $Z_{۰۳} = ۱$ و $Z_{۰۱} = Z_{۰۲} = ۰$. از طرفی، ژن مربوط به گره غیرهاب ۲ شامل سه صفر است. پس عدد صحیح ۰ از بخش دوم آن استخراج می‌شود:

$$۰ \overset{n_{hub}}{\equiv} ? \qquad \Longrightarrow \qquad ۰ \overset{۲}{\equiv} ۰$$

که رابطه فوق بیانگر تخصیص این گره به نزدیکترین هاب یعنی گره ۱ است. بعبارتی، $Z_{۲۱} = ۱$ و $Z_{۲۰} = Z_{۲۳} = ۰$.

۳-۱-۲ جستجوهای محلی ابتکاری HeurLoc و HeurAlloc

در این بخش، جواب هایی که توسط الگوریتم تکاملی تولید شده‌اند با دو جستجوی محلی ابتکاری HeurLoc و HeurAl-loc بهبود می‌یابند. بیان این نکته ضروری است که این دو جستجوگر صرفاً برای افرادی از جامعه اعمال می‌گردند که دارای

اثر دوگانه زیر باشند:

- تنوع ژنتیکی جامعه را به طور قابل ملاحظه‌ای کاهش نمی‌دهند؛
- زمان اجرای الگوریتم را بیش از اندازه افزایش نمی‌دهند.

تجزیه و تحلیل و مقایسه تعداد هاب‌های ایجادشده در جواب‌های بهینه و بهترین جواب‌های بدست آمده از الگوریتم تکاملی نشان می‌دهد که در همه‌ی موارد آزمایشی، الگوریتم موردنظر همان تعداد هاب‌های تعیین مکان شده در جواب‌های بهینه مربوطه را برمی‌گرداند. بنابراین، تعداد بهینه هاب‌ها به‌راحتی مشخص می‌شود. بعد از تعیین تعداد هاب‌ها از بخش تکاملی الگوریتم، دو جستجوی محلی کارآمد به‌کاربرده می‌شوند که به تعیین مکان‌های بهینه هاب‌ها و تخصیص گره‌های غیرهاب کمک زیادی می‌کنند. ابتدا روش HeurLoc به‌کار می‌رود که به دنبال معاوضاتی بین گره‌های مجموعه هاب داده شده و دیگر گره‌های موجود در شبکه می‌گردد که در نهایت منجر به یکسری بهبود شوند. در ادامه، روش HeurAlloc به ارزیابی تمام تخصیص‌های مجدد هر گره غیرهاب به یکی از هاب‌های تعیین مکان شده می‌پردازد. برای راحتی کار نمادهای مهم مورد استفاده در این بخش را در جدول ۳-۱ معرفی می‌کنیم.

جدول ۳-۱: معرفی پارامترها

نماد	توضیح
s	جستجوی محلی HeurLoc برای هر s امین فرد اجرا می‌شود
q	هنگامیکه جستجوگر HeurLoc سعی در معاوضه یک هاب به جای یک غیرهاب دارد فقط به جستجوی q تا از نزدیکترین گره‌های غیرهاب می‌پردازد
r	جستجوی محلی HeurAlloc برای هر r امین فرد اجرا می‌شود
N, G	افراد
$\bar{G}$	کد ژنتیک فرد G
$MA_Obj(N)$	مقدار تابع هدف فرد N
$H(\bar{G})$	مجموعه گره‌های هاب مشخص شده از کد ژنتیک $\bar{G}$ فرد G
$NH(\bar{G})$	مجموعه گره‌های غیرهاب مشخص شده از کد ژنتیک $\bar{G}$ فرد G
nh	آرایه‌ای موقتی که برای یک گره هاب معین از مجموعه $H(\bar{G})$ برخی افراد G تشکیل شده و شامل گره‌هایی از مجموعه $NH(\bar{G})$ است که برمبنای هزینه‌های حمل‌ونقل از آن گره هاب به‌صورت صعودی مرتب شده‌اند
$sum(G, k)$	مجموع هزینه‌های حمل‌ونقل از گره غیرهاب k از طریق هاب تخصیص داده شده به آن که برای یک فرد G محاسبه می‌شود

حال به تبیین جزئیات روش HeurLoc می‌پردازیم. قابل ذکر است که در طی این فرآیند تعداد گره‌های هاب بدون تغییر باقی می‌مانند و جستجوی محلی در روندی تکراری برای یافتن انتخابی بهتر برای مکان‌های هاب به‌کار برده می‌شود.

از آنجایی که فراخوانی‌های ارزیابی تابع هدف جزء زمان‌برترین بخش الگوریتم به‌حساب می‌آیند، به منظور حفظ کارایی جستجوگر HeurLoc اقدامی جهت کاهش این فراخوانی‌ها صورت خواهد گرفت به نحوی‌که فضای جستجو به‌طور قابل ملاحظه‌ای محدودیت پیدا نکند. در این راستا جستجو به درصد معینی از افراد جامعه محدود خواهد شد که برای دستیابی به این هدف پارامتر s معرفی شده است. در واقع، جستجوی محلی فقط برای هر sامین فرد اعمال می‌گردد. بااستفاده از استراتژی مرتب‌سازی آرایه‌ای گره‌های غیرهاب برای هر گره هاب تأسیس شده، جستجو به مناطق امید دهنده هدایت می‌شود. ابتدا مجموعه گره‌های هاب $H(\bar{G})$ و مجموعه گره‌های غیرهاب $NH(\bar{G})$ از کد ژنتیکی $\bar{G}$ یک فرد G بدست می‌آیند. برای هر گره هاب $k \in H(\bar{G})$ اعضای مجموعه $NH(\bar{G})$ با توجه به هزینه حمل‌ونقل بین این اعضا و گره هاب k، داخل آرایه nh به‌صورت صعودی مرتب می‌شوند. نهایتاً با انجام فرآیند روی اندیس i سعی می‌شود تا گره هاب k با گره غیرهاب $nh[i]$ معاوضه شود. از آنجایی که هیچ محدودیت ظرفیتی وجود ندارد احتمال معاوضه یک گره هاب با گره غیرهابی که در فاصله‌ای دور قرار گرفته بسیار کم است. بنابراین لزومی برای هدایت جستجو به اعماق آرایه nh دیده نمی‌شود و تنها q تا از اولین گره‌های غیرهاب $nh[i]$ که $i = ۱, \cdots, q$ با گره هاب جاری $k \in H(\bar{G})$ قابل معاوضه خواهند بود. با محدود کردن جستجو از آرایه nh به q تا از اولین اعضای آن، از زمانبر بودن و دشوارشدن کنترل و سنجیدن الگوریتم ممانعت بعمل می‌آید. بعد از معاوضه واحد گره غیرهاب $nh[i]$ و گره هاب k همه‌ی گره‌های غیرهاب که قبلاً به k تخصیص داده شده بودند به گره هاب جدید تخصیص دوباره می‌یابند. اکنون گره k که به گره‌ای غیرهاب تبدیل شده است به هاب $nh[i]$ تخصیص داده می‌شود در حالی‌که این گره هاب به‌طورپیش فرض به خودش تخصیص می‌یابد.

بعد از معاوضه هاب و انجام تخصیص‌های مجدد ضروری می‌توان مقدار جدید تابع هدف را محاسبه نمود. تنها در صورتیکه مقدار جدید از مقدار قدیمی بهتر باشد تغییرات لازم در کد ژنتیک فرد انجام می‌شود و فرد بهبود یافته جدید بدست می‌آید. روند شرح داده شده تا زمان حصول اولین بهبود روی یک فرد تکرار می‌شود که به اجتناب از همگرایی نابهنگام کمک می‌کند. بهبودهای متوالی متعدد برای تعدادی از افراد منجر به فقدان تنوع در کد ژنتیکی و به‌طور مؤثرتر باعث همگرایی به یک بهینه محلی می‌شود. فرد بهبود یافته ممکن است در نسل‌های بعدی بیشتر بهبود یابد. ساختار اصلی جستجوی محلی HeurLoc درالگوریتم ۳-۱ ارائه شده‌است.

بعد از بهبود مکان‌های انتخاب شده برای تأسیس هاب‌ها، جستجوی محلی HeurAlloc روی بخش تخصیص مسئله متمرکز می‌شود. در این بخش، مجموع هزینه‌های ثابت برای مکانیابی هاب‌ها بدلیل عدم تغییر هاب‌های تعیین مکان شده به‌صورت قبلی باقی می‌مانند. با استفاده از این جستجوگر، حرکت‌های ساده اما مؤثری جهت جستجوی همسایگی یک جواب داده شده G صورت خواهد گرفت. توپولوژی همسایگی جواب به این صورت است که برای هر گره غیرهاب، انتساب این گره از هابی که اکنون به آن اختصاص دارد به یک گره هاب دیگر تغییر می‌کند و در نتیجه قانون " تخصیص نزدیکترین هاب " در مورد تخصیص گره‌های غیرهاب به گره‌های هاب نادیده گرفته خواهدشد. در بسیاری از جواب‌های بهینه دیده می‌شود که گره‌های غیرهاب لزوماً به نزدیکترین هابشان تخصیص نمی‌یابند. با توجه به این‌که جستجوی محلی موردنظر درگیر فراخوانی‌های زیاد ارزیابی تابع هدف MA_Obj به‌عنوان زمانبرترین بخش الگوریتم است این روش نیز تنها روی درصد معینی از جمعیت اعمال می‌شود. بدین منظور پارامتر r معرفی می‌گردد و هریک از r امین افراد در معرض جستجو

الگوریتم ۳-۱ ساختار جستجوی محلی HeurLoc

ورودی: جواب تولید شده توسط الگوریتم تکاملی و پارامترهای s و q ،
خروجی: بهبود مکان تأسیس گره‌های هاب.

۱: هر فرد G در جواب را در نظر بگیرید.
۲: اگر اندیس فرد G به پیمانه s دارای ارزش ۰ بود
گره‌های غیرهاب مشخص شده از کدژنتیکی $\bar{G}$ آن فرد را در مجموعه $NH(\bar{G})$ قرار دهید.
۳: برای هر گره هاب k از مجموعه هاب‌های تعیین مکان شده $H(\bar{G})$ با توجه به هزینه حمل و نقل بین آن گره و گره‌های غیر هاب، اعضای $NH(\bar{G})$ را در آرایه nh به‌صورت صعودی مرتب کنید.
۴: برای $i = ۱ : q$ مراحل زیر را انجام دهید.
۵: یک کپی از فرد G را در N قرار دهید.
۶: سعی کنید گره غیرهاب $nh[i]$ را با گره هاب k معاوضه کنید.
۷: در فرد N گره غیرهاب $nh[i]$ را به‌عنوان یک هاب، گره هاب k را به‌عنوان یک غیرهاب، تخصیص هاب جدید $nh[i]$ را به خودش، تخصیص گره غیرهاب k را به $nh[i]$ و تخصیص مجدد همه گره‌های غیرهاب از k به $nh[i]$ را لحاظ کنید.
۸: $MA_Obj(N)$ را محاسبه کنید.
۹: اگر رابطه $MA_Obj(G) > MA_Obj(N)$ برقرار بود فرد N را با فرد G جایگزین کنید.
۱۰: $MA_Obj(N)$ را به جای $MA_Obj(G)$ قرار دهید.

قرار خواهند گرفت. از فواید این محدودیت جستجو می‌توان به حفظ کارایی روش ابتکاری و در کل به مقابله با طولانی شدن قابل توجه زمان اجرای الگوریتم ممتیک اشاره کرد.

جستجوگر HeurAlloc هر گره غیرهاب k در جواب فعلی را در نظر می‌گیرد و سعی در تخصیص مجدد آن به گره‌های دیگر مجموعه هاب‌های تأسیس شده دارد. در پرتو این تلاش است که جواب جدید N تولید می‌شود. برای هر جواب داده شده G فرض بگیرید که $sum(G,k)$ بیانگر مجموع هزینه‌های حمل‌ونقل برای ترافیک جریانی است که از گره مبدأ k به تمام گره‌های مقصد در شبکه روانه می‌شود. در صورتی‌که $sum(N,k)$ که بعد از تخصیص دوباره گره k به یک هاب متفاوت بدست می‌آید کمتر از $sum(G,k)$ باشد آنگاه مقدار تابع هدف جدید فقط برای فرد N محاسبه و با مقدار تابع هدف G مقایسه می‌گردد.توجه داشته باشید که به احتمال زیاد برای نمونه‌های سایز بزرگ و یا بعبارتی نمونه‌های دارای تعداد زیادی گره حتی اگر نامساوی $sum(G,k) > sum(N,k)$ برقرار باشد لزوماً در کل هزینه‌های حمل‌ونقل کاهشی رخ نخواهد داد.

فرض کنید G یک جواب و k یک گره غیرهاب $(k \in NH(\bar{G}))$ و i و j گره‌هایی هاب $(i,j \in H(\bar{G}))$ باشند و k به j تخصیص داده شده باشد. با تخصیص مجدد گره k به گره هاب i و تولید جواب جدید N، تمامی مکان‌ها و تخصیص‌های دیگر به همان صورتی‌که در G قرار داشتند در N دست نخورده باقی می‌مانند. نهایتاً فرض کنید $sum(G,k) > sum(N,k)$، که به معنی کاهش مجموع هزینه‌های حمل‌ونقل ترافیک روانه شده از k توسط هاب i در N در مقایسه با ترافیک روانه شده از k توسط هاب j در G است. با این حال در راه‌حل N همه ترافیک جریان یافته در

الگوریتم ۳-۲ ساختار جستجوی محلی HeurAlloc

ورودی: جواب تولید شده توسط الگوریتم تکاملی و پارامتر r،
خروجی: بهبود تخصیص گره‌های غیر هاب .

۱: هر فرد G در جواب را در نظر بگیرید.
۲: اگر اندیس فرد G به پیمانه r دارای ارزش ۰ بود
مجموعه $NH(\bar{G})$ شامل گره‌های غیرهاب مشخص شده از کدژنتیکی $\bar{G}$ آن فرد را تشکیل دهید.
۳: $Sum(G,k)$ را که نشان دهنده مجموع هزینه‌های حمل‌ونقل از گره غیرهاب k بواسطه گره هاب j تخصیص داده شده به آن است را برای هر $k \in NH(\bar{G})$ محاسبه کنید.
۴: برای هر گره هاب $i \in H(\bar{G})$ که $i \neq j$ مراحل زیر را انجام دهید.
۵: یک کپی از فرد G را در N قرار دهید.
۶: برای تخصیص مجدد گره غیرهاب k به یک گره هاب دیگر $i \in H(\bar{G})$ تلاش کنید.
۷: تخصیص مجدد k به i را در N لحاظ کنید.
۸: $Sum(N,k)$ را محاسبه کنید.
۹: اگر رابطه $Sum(G,k) > Sum(N,k)$ برقرار بود آنگاه به محاسبه $MA_Obj(N)$ بپردازید.
۱۰: در صورت برقراری رابطه $MA_Obj(G) > MA_Obj(N)$ می‌توانید مقدار $MA_Obj(N)$ را به جای مقدار $MA_Obj(G)$ جایگزین کنید.

جهت دیگر از برخی گره‌های غیرهاب مانند l به گره غیرهاب k توسط هاب i و نه از طریق هاب j تعیین مسیر خواهند شد که ممکن است باعث عدم برقراری نامساوی $sum(G,l) > sum(N,l)$ شود. به همین دلیل، صرفاً N به‌عنوان یک نامزد برای جواب جدید مطرح است و در ادامه باید به بررسی بهبود کل مقدار تابع هدف بعد از تغییر تخصیص گره k پرداخت و فقط در صورت بروز بهبود است که جواب جدید بهبود یافته N حفظ شده و جواب قبلی G کنار گذاشته می‌شود. طرح اولیه جستجوی محلی HeurAlloc درالگوریتم ۳-۲ آمده است.

فهرست منابع

[۱] یقینی، مسعود و اخوان کاظم زاده، محمد رحیم. الگوریتم‌های بهینه‌سازی فراابتکاری. انتشارات جهاددانشگاهی (واحد صنعتی امیرکبیر)، تهران، ویرایش چاپ سوم، ۱۳۹۰.

[2] Silva, M. R. and Cunha, C. B. New simple and efficient heuristics for the uncapacitated single allocation hub location problem. *Comput. Oper. Res.*, 36(12):3152–3165, 2009.

[3] Laporte, G., Nickel, S., and Saldanha da Gama, Francisco. *Location Science*. Springer International Publishing, Springer Cham Heidelberg New York Dordrecht London, 2015.

[4] Toh, R. S. and Higgins, R. C. The impact of hub and spoke network centralization and route monopoly on domestic airline profitability. *Tansportation*, 24(4):16–27, 1985.

[5] Hakimi, S. L. Optimum locations of switching centers and the absolute centers and medians of a graph. *Oper. Res.*, 12(3):450–459, 1964.

[6] O'Kelly, M. E. The location of interacting hub facilitie. *Tansportation Science*, 20(2):92–106, 1986a.

[7] O'Kelly, M. E. Activity levels at hub facilities in interacting networks. *Geographical Analysis*, 18(4):343–356, 1986b.

[8] O'Kelly, M. E. A quadratic integer program for the location of interacting hub facilities. *European Journal of Operational Research*, 32(3):393–404, 1987.

[9] O'Kelly, M. E. and Miller, H. The hub network design problem: A review and synthesis. *Transport Geography*, 2(1):31–40, 1994.

[10] Campbell, J. F. A survey of network hub location. *Studies in Locational Analysis*, 6(6):31–49, 1994a.

[11] Klincewicz, J. G. Hub location in backbone/tributary network design: a review. *Location Science*, 6(1-4):307–335, 1998.

[12] Bryan, D. L and O'Kelly, M. E. Hub-and-spoke networks in air transportation: an analytical review. *regional science*, 39(2):275–295, 1999.

[13] Campbell, J. F., Ernst, A. T., and Krishnamoorthy, M. Hub location problems. *Facility location: applications and theory*, 1:373–407, 2002.

[14] Alumur, Sibel and Kara, Bahar Y. Network hub location problems: The state of the art. *European journal of operational research*, 190(1):1–21, 2008.

[15] Campbell, J. F. and O'Kelly, M. E. Twenty-five years of hub location research. *Transportation Science*, 46(2):153–169, 2012.

[16] Farahani, R., Hekmatfar, M., Boloori Arabani, A., and Nikbakhsh, E. Hub location problems: A review of models, classification, solution techniques, and applications. *Comput. Ind. Eng.*, 64(4):1096–1109, 2013.

[17] Farahani, R.Z. and Hekmatfar, M. *Facility Location: Concepts, Models, Algorithms and Case Studies*. Physica-Verlag HD, 2009.

[18] Campbell, J. F. Integer programming formulations of discrete hub location problems. *European Journal of Operational Research*, 72(2):387–405, 1994.

[19] Campbell, J. F. Hub location problems and the p-hub median problem. center for business and industrial studies, university of missouri – st. louis, st. louis, mo. 1991.

[20] Costa, M. G., Captivo, M. E., and Climaco, J. Capacitated single allocation hub location problem-a bi-criteria approach. *Comput. Oper. Res.*, 35(11):3671–3695, 2008.

[21] Marić, M., Stanimirović, Z., and Stanojević, P. An efficient memetic algorithm for the uncapacitated single allocation hub location problem. *Soft Comput.*, 17(3):445–466, 2013.

[۲۲] شاهی، سمیرا. حل مسائل مکانیابی با استفاده از روش‌های فراابتکاری. پایان‌نامه کارشناسی‌ارشد، دانشکده ریاضی، دانشگاه حکیم سبزواری، شهریور ۱۳۹۲.

[23] Glover, F. Future paths for integer programming and links to artificial intelligence. *Comput. Oper. Res.*, 13(5):533–549, 1986.

[۲۴] عشقی، کورش و کریمی نسب، مهدی. بهینه‌سازی ترکیبی و الگوریتم‌های فراابتکاری. کلک زرین، ۱۳۹۱.

[25] Glover, F. and Laguna, M. *Tabu Search*. Kluwer Academic Publishers, Norwell, MA, USA, 1997.

[26] Abyazi Sani, Roya and Ghanbari, Reza. An efficient tabu search for solving the uncapacitated single allocation hub location problem. *Comput. Ind. Eng.*, 93(c):99–109, 2016.

[27] Sun, Mingha. Solving the uncapacitated facility location problem using tabu search. *Comput. Ind. Eng.*, 33(9):2563–2589, 2006.

[28] Burke, E.K. and Kendall, G. *Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques*. Springer US, New York, 2 ed. , 2014.

[29] Topcuoglu, H., Corut, F., Ermis, M., and Yilmaz, G. Solving the uncapacitated hub location problem using genetic algorithms. *Comput. Oper. Res.*, 32(4):967–984, 2005.

Hakim Sabzevari University

An Outline of MSc. Thesis

Surname:Hasany pezhman	Name:Fereshteh	Student No.:9313133040

Supervisor: Dr. Mahmood Amintoosi

Advisor: Dr. Mehdi Zaferanieh

Faculty of Mathematics and Computer Science

Program: Applied Mathematics Field:Operational Research

Title of thesis: Solving the uncapacitated single allocation hub location problem using memetic algorithms

Keywords:Memetic Algorithms، Network Design، Hub Location Problems، Local Search

Abstract: In the past decades, Location Science has become a very active research area, attracting the attention of many researchers and practitioners. Hub Location Problems (HLPs) Which are a very practical subset of facility location problems, lie at the heart of network design planning in transportation and telecommunication systems. Most of hub location problems are proven to be NP-hard.Therefore, in the literature, many meta-heuristic algorithms have been used to solve them. In recent years, the combination of meta-heuristic algorithms has become verey common-place such that the best results for real-world optimization problems or classical issues are derived from hybrid algorithms. Memetic algorithms are population-based metaheuristics composed of an evolutionary framework and a set of local search algorithms which are activated within the generation cycle of the external framework.

In this dissertation, an efficient and effective memetic algorithm for solving a variety of hub location problem called uncapacitated single allocation hub location problem (USAHLP) has been investigated.

Hakim Sabzevari University
Faculty of Mathematics and Computer Science

A Thesis Submitted in Partial Fulfilment of the Requirement for the Degree of Master of Science in Applied Mathematics

Solving the uncapacitated single allocation hub location problem using memetic algorithms

Supervisor:
Dr. Mahmood Amintoosi

Advisor:
Dr. Mehdi Zaferanieh

By:
Fereshteh Hasany pezhman

August 2017