دانشکده : ریاضی

گروه : کاربردی

مساله زمانبندی و مکانیابی توام مدل تک ماشینی

دانشجو : طاهره سیار

استاد راهنما :

آقای دکتر جعفر فتحعلی

استاد مشاور:

آقای دکتر نادر جعفری راد

پایان نامه ارشد جهت اخذ درجه کارشناسی ارشد

تیر ماه ۱۳۹۰

چکیده

در دهه های اخیر توجهات زیادی به علمی شدن و بهینه سازی کارها و مسائل کاربردی شده است و دانشمندان سعی در حل مسائل واقعی و پیرامون زندگی دارند. بسیاری از این مسائل و بهینه سازی ها توسط علم تحقیق در عملیات قابل حل هستند که دو شاخه گسترده و مهم آن زمانبندی و مکانیابی می باشد. زمانبندی، چگونگی تخصیص منابع برای انجام فعالیت ها در طول زمان می باشد. و کاربردهای زیادی در صنعت و تکنولوژی دارد که می توان به زمانبندی خط تولید کارخانه ها، زمانبندی حرکت قطارها و... اشاره نمود. مکانیابی نیز پیدا کردن مکانی جدید در فضای مورد نظر است به گونه ای که فاصله آن تا نقاط موجود کمینه گردد. مکانیابی خود، شاخه های گوناگونی دارد و کاربردهای فراوانی در زندگی واقعی دارد، از جمله کاربردهای آن می توان به مکانیابی محل استقرار کارخانه، مراکز اورژانس، ایستگاههای آتش نشانی، پایانه های حمل و نقل و... اشاره نمود.
بحثی که در این پایان نامه مورد بررسی قرار گرفته است، مساله مکانیابی و زمانبندی توام مدل تک ماشینی می باشد. این مساله اولین بار در سال ۲۰۰۲ مطرح گردید و تاکنون تنها حالت تک ماشینی آن بحث و بررسی شده است. حالت های دیگر آن مثل حالت چند ماشینی بسیار مشکل و پیچیده است و تا به حال کسی آن را مورد بررسی قرار نداده است.

ما در فصل نخست پایان نامه دو نوع خاص از زمانبندی تک ماشینی را بررسی می کنیم. در فصل دوم مکانیابی تک وسیله ای با دو تابع هدف کمترین مجموع و مینیماکس و در فصل سوم مساله توام مکانیابی و زمانبندی تک ماشینی را روی صفحه مورد مطالعه قرار می دهیم. در فصل چهارم مکانیابی تک وسیله ای بر روی خط، به گونه ای که نقاط تقاضا بر روی صفحه و نقطه جدید روی خط مشخص قرار بگیرد را بررسی می کنیم. در فصل پنجم مساله مکانیابی و زمانبندی توام مدل تک ماشینی بر روی خط را مطرح و الگوریتمی برای آن ارائه می دهیم. در فصل ششم روش مثلث بزرگ مثلث کوچک و در نهایت در فصل هفتم مساله مکانیابی و زمانبندی توام مدل تک ماشینی بر روی شبکه را بررسی می کنیم. در مساله مکانیابی و زمانبندی توام تک ماشینی حالت های زیادی برای بررسی وجود دارد، به عنوان مثال می توان این مساله را در فضای سه بعدی مورد بررسی قرار داد. همچنین این مساله بر روی صفحه را می توان با نرم بلوکی حل کرد و الگوریتمی برای حل آن ارائه داد. چون مساله مکانیابی و زمانبندی توام، مساله جدیدی است، مسائل زیادی از آن مورد بررسی قرار نگرفته است که می توان به آن ها پرداخت.

لیست مقالات مستخرج از پایان نامه

[۱] طاهره سیار، جعفر فتحعلی،" مسئله زمانبندی ومکانیابی توام مدل تک ماشینی روی خط با تابع هدف مینیماکس" (۱۳۸۸)،

[۲] طاهره سیار، جعفر فتحعلی،" مسئله زمانبندی ومکانیابی توام مدل تک ماشینی روی خط". (۱۳۸۹)

[۳] طاهره سیار، جعفر فتحعلی،" مسئله زمانبندی ومکانیابی توام مدل تک ماشینی در فضای سه بعدی"(۱۳۹۰)

فهرست مندرجات

فهرست اشکال

فهرست جداول

فصل ۱
زمانبندی تک ماشینی

۱-۱ مقدمه

یک عامل موثر برای انجام بهتر و مدیریت مناسب تر فعالیت های مختلف، استفاده از الگـویی اسـت که اختصاص منابع مورد نیاز به این فعالیت ها را به بهترین وجه ممکن انجام دهد. از مسائلی که ما را در این زمینه یاری می دهد **مساله زمانبندی**[۱] می باشد که عبارت است از چگـونگی تخصـیص منابع برای انجام فعالیت ها در طول زمان. مسائل زمانبندی در صنعت کاربرد فـراوان دارنـد، بـدین جهت امروزه به آن ها اهمیت بسیار داده می شود.

مساله زمانبندی اولین بار در سال ۱۹۳۶ برای بررسی جدول زمانی راه آهن بـه کـار بـرده شـد[۳]. سپس دراوایل سال ۱۹۵۰ این مساله توجه بسیاری از دانشمندان را به خود جلب کرد. از آن زمـان به بعد صورت هایی از این مساله همراه با فرضیات مختلف بیان شد و دانشمندان زیادی در مورد آن به بحث و ارائه راه حل پرداختند به گونه ای که امروزه این مساله به صورت یک مبحث گسترده در آمده است که کار در این زمینه مطالعات خاصی را می طلبد.

۱-۲ تعریف و اهداف مسائل زمانبندی

در اغلب مسائل زمانبندی یک مجموعه از منابع یا سرویس دهنده ها و یک مجموعه از مصـرف کننده ها را داریم که باید سرویس دهنده ها به مصرف کننده ها به نحوی اختصاص یابند که یکـی از معیار های عملکرد بهینه شود. سرویس دهنده ها می توانند یک منبع، یک **پردازش گر**[۲] ، یـک کارمند، یک ماشین و... باشند و مصرف کننده می تواند یک برنامه کامپیوتری، یـک مشـتری، یـک **کار**[۳] و ... باشد. در این پایان نامه از کلمه ماشین به جای کلمه سرویس دهنـده و از کلمـه کـار بـه جای کلمه مصرف کننده استفاده می کنیم.

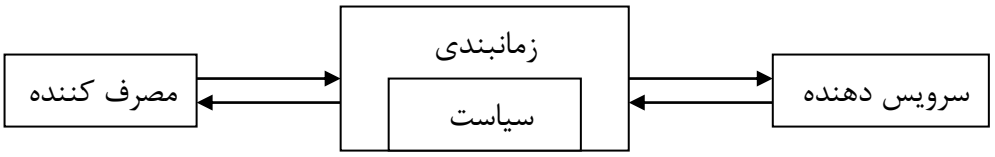
معیار عملکرد رابطه مستقیم با اهداف مساله دارد و در حقیقت همان تابع هدف مساله می باشد که در اغلب مسائل به صورت حداقل کردن زمان ظاهر می شود. همچنین تابع هدف ممکـن اسـت حداقل کردن زمان کل سرویس، حداقل کردن میانگین زمان انتظار کارها بـرای دریافـت سـرویس، بهینه کردن پیچیدگی زمانی و ... باشد.

جهت اجرای کارها توسط پردازش گر، در یک مساله زمانبندی سیاستی اعمال می شود که تابع هدف بهینه شود. یک سیاست زمانبندی بهینـه بایـد دو عمـل چگـونگی ترتیـب پـردازش کارهـا و تخصیص پردازشگرها به کارها را مشخص کند. فرم کلی سیستم زمانبندی به صـورت شـکل (۱-۱) [۱] می باشد.

---

[1] Scheduling Problem
[2] Processor
[3] Job

دو مشخصه جهت مقایسه و سنجش زمانبندی های مختلف، عملکرد و کارایی آن ها مـی باشـد. عملکرد مشخص می کند که سیاست تا چه حد به بهینگی نزدیک است، که هر چـه عمکـرد بهتـر باشد سیاست بهتر است و عملکرد بهینه سیاست بهینه را در بر دارد. کارایی نیز به این معنی اسـت که سیاست زمانبندی تمام حالت های مساله را جواب گو باشد.



شکل ۱-۱ سیستم زمانبندی

۱-۳ دسته بندی مسائل زمانبندی

هر مساله زمانبندی فرضیاتی دارد که بر اساس آن ها سیاست زمانبندی طرح ریزی مـی شـود. اعمال فرض های مختلف، مسائل زمانبندی گوناگونی به وجـود مـی آورد. بعضـی از فرضـیاتی کـه موجب ایجاد مسائل زمانبندی مختلف می شوند در ادامه آمده است:

۱- آیا تمام کارها در ابتدا در سیستم موجودند یا اینکه به تدریج وارد سیسـتم مـی شـوند. کـه در حالت اول ، زمانبندی مساله به صورت **ایستا**[۴] و در حالت دوم، زمانبندی به صورت **پویا**[۵] انجام می شود. در حالت پویا زمان ورود کارها ممکن است به صورت یک تابع احتمالی باشد.

۲- آیا قطع پردازش یک کار مجاز است یا خیر. یعنی آیا می توان در حین انجام یک کـار پـردازش آن را متوقف نمود و ادامه کار را در زمان دیگری انجام داد، که بسته به مورد، زمانبندی **قابل قطع**[۶] یا **غیر قابل قطع**[۷] نامند.

۳- اگر تعداد پردازش گر ها یکی باشد، زمانبندی را **تک ماشـینی**[۸] و اگـر چنـد تـا باشـد، **چنـد ماشینی**[۹] گویند.

---

[4] Static
[5] Dynamic
[6] Preemptive
[7] Non preemptive
[8] Single Machine
[9] Multiple Machine

۴- جریان پردازش کارها توسط پردازشگرها چگونه است. یعنی هر کار جهت پردازش شدن به چه پردازشگرهایی نیاز دارد. اگر هر کار دارای جریان پردازش مخصوص به خود باشد، مساله را **کارگاهی**[۱۰] نامیده و اگر تمام کارها دارای جریان پردازش یکسان باشند، مساله را **کارگاهی خاص**[۱۱] می نامند. همچنین اگر ترتیب انجام پردازش توسط پردازشگرهای مختلف روی کارها مهم نباشد، مساله را **کارگاهی بدون شرط** [۱۲] نامند، و در صورتی که در یک مساله کارگاهی خاص ترتیب پردازش کارها بر روی ماشین ها یکسان باشد، مساله را **کارگاهی خاص جایگشتی**[۱۳] نامند.

۵- پردازشگرها می توانند به طور **موازی** [۱۴] باشند، یعنی پردازش هر کار می تواند توسط هر یک از پردازشگرها انجام شود.

۶- آیا کارها از یکدیگر مستقل هستند یا به هم وابسته اند. و شروع پردازش بعضی منوط به پایان پردازش بعضی کارهای دیگر است، که در این حالت زمانبندی همراه با **تقدم**[۱۵] کارها را خواهیم داشت.
۷- هر کار دارای **زمان آماده سازی**[۱۶] می باشد یا نه. اگر کارها دارای زمان آماده سازی هستند، آیا مقدار آن ها از یکدیگر متفاوت است یا با هم برابرند.

مسائل زمانبندی را به طور کلاسیک به صورت $\alpha|\beta|\gamma$ نمایش می دهند. که در آن α معادل است با تعداد پردازش گرها یا ماشین ها، β نشان دهنده فرضیات یا محدودیت هایی است که روی کارهای مساله است که اگر برای β چیزی قید نشود به معنی آن است که هیچ محدویت خاصی برای مساله وجود ندارد. و γ نشان دهنده تابع هدف مساله می باشد که باید مینیمم گردد.

۴-۱ ضوابط بهینگی در مسائل زمانبندی

اگر زمان اتمام کار j ام از مجموعه کارهای $N=\{1,...,n\}$ که قرار است پردازش گردد را با C_j و هزینه مربوطه را با $f_j(C_j)$ نشان دهیم، دو نوع تابع هزینه کلی وجود دارد که به صورت زیر می باشد:

$$f_{\max}(C) := \max\{f_j(C_j) |\ j=1,...,n\} \qquad (۱-۱)$$

---

[10] Job Shop
[11] Flow Shop
[12] Open Shop
[13] Preemptive Flow Shop
[14] Parallel Processor
[15] Precedence
[16] Release date

$$\sum f_j(C) := \sum_{j=1}^{n} f_j(C_j) \qquad (۲-۱)$$

بنابراین مساله ، یافتن دنباله زمانبندی مناسبی است که این توابع را مینیمم کند.

بیشترین تابع هدف هایی که در مسائل زمانبندی استفاده می شود عبارتند از کمینه کردن یکی از توابع زیر: **کل زمان اتمام کار**[۱۷] یا $\max\{C_j \mid j=1,...,n\}$ و **مجموع زمان اتمام کارها** [۱۸] یا $\sum_{j=1}^{n} C_j$ و مجموع زمان اتمام کارهای وزن دار[۱۹] یا $\sum_{j=1}^{n} w_j C_j$ که در آن w_j وزن مر بوط به کار j ام است، می باشد.

بعضی از توابع هدف به **زمان های سررسید**[۲۰] d_j (زمانی که پردازش کار j ام باید به اتمام برسد) که مربوط به کار j ام است بستگی دارد. فرض کنید برای هر کار j ، **دیرشدگی**[۲۱] به صورت $L_j := C_j - d_j$، **دیرکرد**[۲۲] به صورت $T_j = \max(0, C_j - d_j)$ و یکای جریمه به صورت زیر تعریف شود:

$$U_j := \begin{cases} 0 & if\ C_j < d_j \\ 1 & otherwise \end{cases}$$

تابع هدف های گوناگون دیگری که برای مسائل زمانبندی تعریف می شوند عبارتند از: کمینه کردن ماکزیمم تاخیر یا $L_{\max} := \max_{j=1}^{n} L_j$ ، کمینه کردن مجموع زمان های دیر کرد کارها $\sum T_j$ ، کمینه کردن مجموع وزن دار زمان های دیر کرد کارها $\sum w_j T_j$ ، $\sum U_j$ ، $\sum w_j U_j$ و ... .

۱-۵ مساله زمانبندی تک ماشینی

مسائل زمانبندی تک ماشینی ساده ترین نوع مسائل زمانبندی است. اما با وجود سادگی حالت تک ماشینی بسیار مهم است و سنگ بنای مسائل زمانبندی است. مساله زمانبندی تک ماشینی اولین بار تحت پژوهش های گسترده جکسون[۲۳] [۴] و اسمیت[۲۴] [۵] بدست آمد. در حالت کلی

---

[17] Makespan
[18] Total completion time
[19] Weighted total completion time
[20] Due dates
[21] lateness
[22] tardiness
[23] Jackson
[24] Smith

مساله زمانبندی تک ماشینی، زمانبندی یک مجموعه $N=\{1,...,n\}$ از n کار روی یک ماشین می باشد که این ماشین در آن واحد می تواند حداکثر یک کار را پردازش کند.

در مسائل زمانبندی تک ماشینی غیر قابل قطع برای بدست آوردن توابع هدف، کافیست که دنباله ای بهینه از کارها را بیابیم تا جواب را توصیف کند.

معیارها و فرضیات گوناگون در مسائل زمانبندی تک ماشینی می تواند حالت های مختلفی از مسائل زمانبندی تک ماشینی را به وجود آورد، که هر یک از آن ها دارای پیچیدگی های مختلف می باشند. در دو زیر بخش بعدی به بررسی دو نوع خاص از مسائل زمانبندی تک ماشینی می پردازیم که یکی از آن ها مساله زمانبندی تک ماشینی با زمان های آماده سازی و تابع هدف مینیمم کردن کل زمان اتمام کارها است که به طور کلاسیک به صورت $1|r_j|C_{\max}$ نمایش داده می شود و دیگری مساله زمانبندی تک ماشینی با زمان های آماده سازی و تابع هدف مینیمم کردن مجموع زمان اتمام کارهاست که به صورت $1\left|r_j\right|\sum_{j=2}^{n}C_j$ نشان داده می شود.

۱-۵-۱ بررسی مساله $1|r_j|C_{\max}$

فرض کنید که مجموعه $N=\{1,...,n\}$ از کارها وجود دارد. هر کار j دارای زمان پردازش نامنفی p_j و زمان آماده سازی r_j می باشد که باید به صورت غیر قابل قطع روی تک ماشین پردازش گردد به گونه ای که کل زمان اتمام کار مینیمم گردد. ماشین در آن واحد می تواند تنها یک کار را پردازش کند. این مساله زمانبندی، مساله ای بسیار ساده در بین مسائل زمانبندی است. برای بدست آوردن جواب بهینه مساله زمانبندی کافیست دنباله ای بهینه از کارها را به دست آوریم.

دنباله کارها که روی ماشین پردازش می شوند، با یک جایگشت π از $\{1,...,n\}$ تعریف می شوند، که $\pi(j)=i$ به این معنی است که کار i، jامین کاری است که پردازش می شود. مجموعه تمام جایگشت های $\{1,...,n\}$ با Π_n نشان داده می شود. به ازای هر دنباله $\pi\in\Pi_n$ و هر مکان ماشین، **زمان کامل شدن**[۲۵] برای هر کار $i\in N$ با استفاده از رابطه بازگشتی زیر تعریف می شود:

$$C_{\pi(1)}=r_{\pi(1)}+p_{\pi(1)}$$
$$C_{\pi(j)}=\max\left\{C_{\pi(j-1)},r_{\pi(j)}\right\}+p_{\pi(j)} \qquad j\in\{2,3,...,n\} \qquad (۳-۱)$$

برای به دست آوردن دنباله بهینه این مساله از قانون زودترین زمان آماده سازی[۲۶] یا (ERD) استفاده می کنیم که در آن، کارها به ترتیب غیر نزولی نسبت به زمان آماده سازی شان مرتب می شوند، یعنی:

---

[25] Completion time

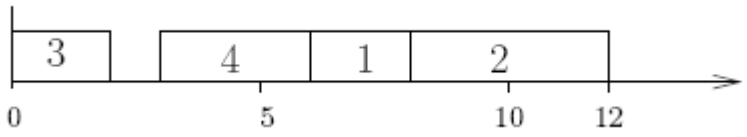
[26] Earliest release date rule

$$r_{\pi(1)} \le r_{\pi(2)} \le \ldots \le r_{\pi(n)} \qquad (۴-۱)$$

برای مرتب کردن این دنباله می توان از الگوریتم های مرتب سازی گوناگون استفاده کرد مثل الگوریتم جستجوی حبابی، دوتایی و ... .

مثال ۱-۱: فرض کنید $N = \{1,2,3,4\}$ مجموعه ای از چهار کار باشد که به ترتیب دارای زمان های پردازش p = (2; 4; 2; 3) و زمان های آماده سازی r = (5; 4; 0; 3) باشند. می خواهیم کارها را به گونه ای زمانبندی کنیم که کل زمان اتمام کار مینیمم گردد.

چون $r_3 = 0 < r_4 = 3 < r_2 = 4 < r_1 = 5$، بنابراین طبق قانون (ERD) دنباله کارها به صورت $\{3,4,2,1\}$ می شود.
نمودار زمانبندی بهینه کارها به صورت زیر است، که در آن مقدار بهینه برای کل زمان اتمام کار برابر است با: $C_{\max} = 12$.



شماره های داخل مستطیل ها شماره کارها ست و نمودار افقی نیز نشان دهنده زمان می باشد.

هنگامی که در مساله تک ماشینی، حق تقدم برای کارها مطرح می شود مساله را به طور کلاسیک به صورت $1|r_j; prec|C_{\max}$ *نمایش می دهند* این مساله مشکل تر است. ولی از آن گونه مسائلی است که دارای الگوریتمی با پیچیدگی زمانی چند جمله ای می باشد، در سال ۱۹۷۳الگوریتمی توسط لاولر[۲۷] با پیچیدگی $O(n^2)$ برای این مساله ارائه شده است[۶].

۱-۵-۲ بررسی مساله $1|r_j|\sum_{j=1}^{n} C_j$

در این بخش مساله مینیمم کردن مجموع زمان اتمام کارها روی تک ماشین با زمان های آماده سازی را بررسی کنیم.
مساله مینیمم کردن مجموع زمان کامل شدن کارها روی تک ماشین هنگامی که زمان های آماده سازی صفر باشد، مساله ای پیش پا افتاده است و می تواند با استفاده از قانون **کوتاه ترین زمان**

[27] Lawler

پردازش[۲۸] یا (SPT) حل گردد. در سال ۱۹۹۷ کارجر[۲۹]و همکارانش [۷] این مساله را حل کردند. ولی هنگامی که زمان های آماده سازی در آن مطرح می شود این مساله، مساله ای NP - سخت است. در اغلب مسائل واقعی، ممکن است همه کارها در زمان صفر قابل دسترس نباشند. از این رو نیاز است که محدویت زمان های آماده سازی مطرح گردد. به دلیل خاصیت NP - سخت بودن مساله الگوریتم هایی که برای حل این مساله ارائه می شود، الگوریتم هایی ابتکاری هستند. و پژوهشگران روی بهبود الگوریتم های تقریبی کار می کنند. شاید اولین کار برجسته روی این مساله توسط فیلیپس[۳۰] و همکاران در سال ۱۹۹۸ [۸] صورت گرفت. الگوریتمی که فیلیپس ارائه کرد، الگوریتم α - زمانبندی نامیده می شد. که در آن کران بالایی برای مجموع زمان اتمام کارها بدست آمده است. کار فیلیپس و همکارانش توسط چکوری[۳۱] و همکارانش در سال ۲۰۰۱ [۹] بهبود یافت، که به موجب آن مقدار کران بالای مجموع زمان اتمام کارها ۱,۵۸ برابر مقدار بهینه مجموع زمان اتمام کارها بود. و الگوریتم بهترین آلفا یا ($BEST\ A$) نامیده شد. الگوریتم دیگری که AEO نامیده می شود توسط اویتونجی[۳۲] و اولولی[۳۳] در سال ۲۰۰۷ [۱۰] پیشنهاد شد.

الگوریتم جدیدی که در این پایان نامه مورد بررسی قرار می گیرد الگوریتم ابتکاری MM نامیده می شود که در سال ۲۰۰۹ توسط ایتونجی[۳۴] و ماساهودو[۳۵] [۱۱] مطرح شده است. در ادامه کارایی این الگوریتم با دو الگوریتم AEO و $HR1$ که توسط اویتونجی و اولولی ارائه شده است، مقایسه خواهد شد. جوابی که این الگوریتم به ما می دهد، جواب بهینه نیست اما تقریب خوبی از جواب بهینه را به ما می دهد.

الگوریتم MM: مبنای الگوریتم این است که اولین کاری که برای زمانبندی انتخاب می شود باید به دقت انتخاب گردد. یعنی اگر اولین کار دارای کمترین زمان پردازش است اما زمان آماده سازی طولانی دارد، می تواند تاثیر خفیفی در مقدار تابع هدف داشته باشد. همچنین اگر دارای کوچکترین زمان آماده سازی باشد لیکن زمان پردازشش طولانی باشد، نیز تاثیر مشابهی خواهد داشت. بنابراین نیاز است که تعادلی بین زمان پردازش و زمان آماده سازی کاری که باید به عنوان اولین کار زمانبندی گردد، وجود داشته باشد. سپس کارهای بعدی می توانند بر طبق ترتیب افزایشی زمان پردازششان زمانبندی گردند. بنابراین الگوریتم ابتکاری MM، کار با کوچکترین مقدار مجموع زمان پردازش و زمان آماده سازی یعنی $(r+p)$ را به عنوان اولین کار و کارهای بعدی را که بر طبق

---

[28] Shortest process time Rule
[29] Karger
[30] Philips
[31] Chekuri
[32] Oyetunji
[33] Oluleye
[34] Oyetunji
[35] Masahudu

ترتیب صعودی زمان پردازششان مرتب شده اند، زمانبندی می کند. مراحل الگوریتم به صورت زیر می با شد.

الگوریتم MM:

گام۱- در ابتدا $J=\{1,2,...,n\}$ را به عنوان کارهای داده شده وJ_S را به عنوان کارهای زمانبندی شده در نظر بگیرید و قرار دهید $k=1$.

گام۲- شاخصه $p+r$ را برای تمام کارهای J محاسبه کنید.

گام۳- اگر $k>1$ به گام ۵ بروید، در غیر اینصورت به گام ۴بروید.

گام۴- کار با کوچکترین شاخصه را از مجموعه کارهای J انتخاب کنید. کار را به عنوان k امین مکان، زمانبندی کنید و این کار را به مجموعه کارهای J_S اضافه کنید و از مجموعه کارهای J حذف کنید.

گام۵- کار با کوچکترین زمان پردازش را از مجموعه کارهای J انتخاب کنید و کار را به عنوان k امین مکان، زمانبندی کنید و به مجموعه کارهای J_S اضافه کنید و از مجموعه کارهای J حذف کنید.

گام۶- اگر مجموعه کارهای J تهی شد، آنگاه به گام ۸ بروید و در غیر این صورت به گام ۷ بروید.

گام۷- k=k+1 و به گام ۵ بروید.

گام۸- زمانبندی مورد نظر با مجموعه کارهای J_S بدست می آید.

گام۹- پایان

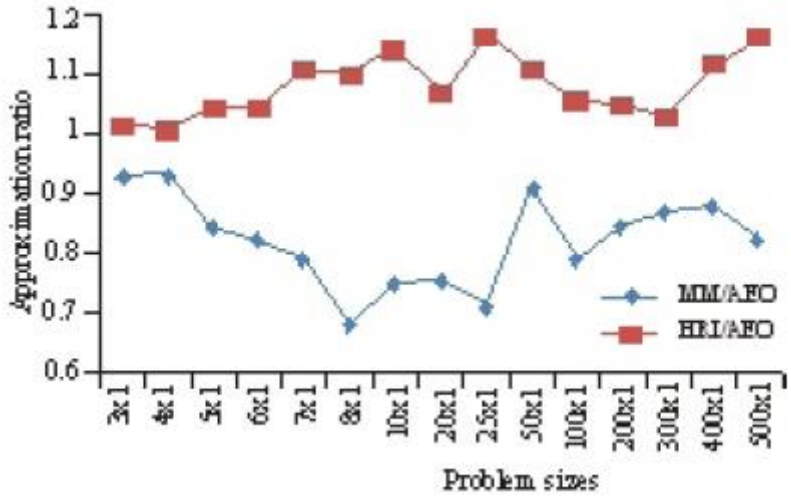
جهت بررسی کارایی الگوریتم MM آن را با دو الگوریتم $HR1$ و AEO مقایسه می کنیم. ایده اصلی الگوریتم AEO، بر مبنای انتخاب کار با کوچکترین زمان پردازش در بین مجموعه کارهایی است که در زمان t قابل دسترس برای پردازش هستند، می باشد و این رویه تا زمانی که تمام کارها پردازش گردند ادامه می یابد. الگوریتم $HR1$ بر این مبنا پایه گذاری شده است که کارها باید طبق ترتیب صعودی مجموع زمان پردازش و زمان آماده سازی کار یعنی (p_i+r_i) زمانبندی شوند.

تحلیل داده ها: برای ارزیابی کارایی الگوریتم ابتکاری پیشنهادی، مسائلی به صورت تصادفی برای زمانبندی روی یک ماشین، تولید شده است که برای ۳ تا ۵۰۰ کار می باشد. زمان پردازش کارها به صورت تصادفی بین ۱-۱۰۰ انتخاب شده است و زمان آماده سازی نیز به صورت تصادفی بین ۰ تا $\left(0.1\times\sum_{j=1}^{n}p_j\right)$ انتخاب می شود.

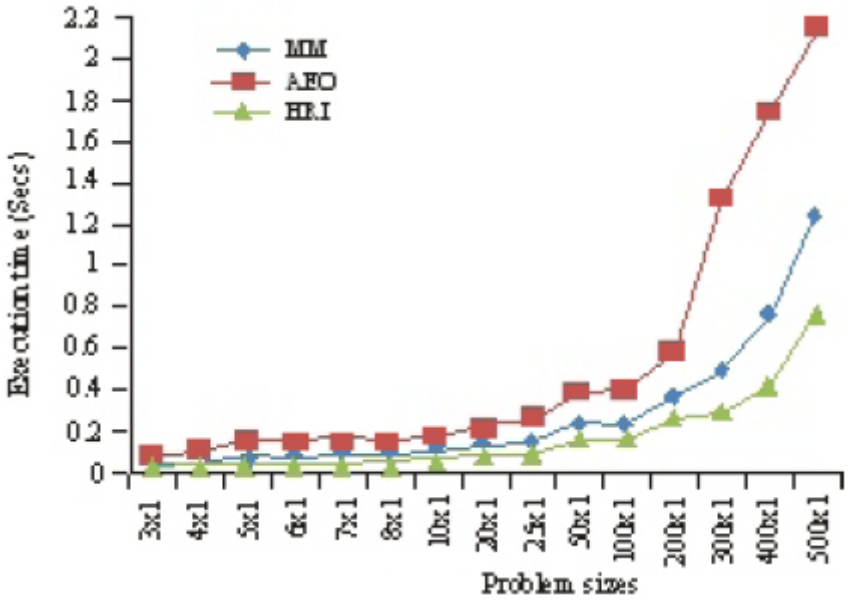
مقادیر میانگین مجموع زمان اتمام کارها از روش های مختلف ، برای اندازه های مختلف مساله در جدول ۱-۱ ([۱۱]) نشان داده شده است. مشاهده می شود که الگوریتم MM کمترین مقادیر مجموع زمان اتمام کارها را به ازای تمام مسائل ارائه می دهد.

جدول ۱-۱ مقادیر میانگین مجموع زمان تکمیل کارها از روش های مختلف حل و برای اندازه مختلف مسائل

	مقادیر میانگین مجموع زمان تکمیل کارها		
اندازه مساله	MM	AEO	$HR1$
3x1	259.40	279.10	283.37
4x1	552.20	593.03	598.90
5x1	614.60	726.53	760.40
6x1	772.77	937.97	980.30
7x1	967.17	1221.93	1355.10
8x1	998.30	1465.67	1616.10
10x1	1777.70	2363.17	2704.87
20x1	6761.03	8947.80	9603.30
25x1	8657.03	12125.00	14152.30
50x1	28597.47	31365.17	34924.97
100x1	29968.30	37859.63	40068.47
200x1	40631.20	47845.60	50268.30
300x1	76801.63	88055.60	90834.73
400x1	86501.65	98155.60	109814.37
500x1	1020835.00	1241789.00	1447865.00



شکل ۱-۲: نسبت تقریبی مجموع زمان تکمیل کارها برای اندازه های مختلف مساله



شکل ۱-۳: مقایسه زمان اجرای اگوریتم ها برای اندازه های مختلف مساله

فصل ۲
مکانیابی تک وسیله ای

۲-۱ مقدمه

مسائل مکانیابی از جمله مسائلی هستند که امروزه کاربردهای فراوانی دارند و توجه بسیاری را به خود جلب کرده اند. پیدایش مسائل مکانیابی را می توان به قرن هفدهم میلادی نسبت داد. یعنی زمانی که فرما مساله زیر را مطرح کرد: " فرض کنید سه نقطه در صفحه داده شده است، نقطه چهارم را به گونه ای بیابید که مجموع فاصله های آن تا سه نقطه داده شده کمینه گردد". اولین تعریف مساله مکانیابی به صورت کاربردی در ۱۹۰۹ م توسط وبر [۱۲] ارائه شد. اما مطالعات جدی بر روی این مساله از زمانی شروع شد که در ۱۹۶۴ حکیمی تابع هدف این مسائل را به صورت کمترین مجموع[۳۶] و مینیماکس[۳۷] مطرح کرد [۱۳].

۲-۱-۱ هدفهای مسایل مکانیابی

مسائل مکانیابی هدفهای مختلفی را دربردارند. هدفها در شناسایی و اولویت بندی معیارهای تصمیم گیری در یک مساله مکانیابی و زیر معیارهای آنها اهمیت و نقش مهمی دارند. در یک تقسیم بندی، هدفهای مسائل مکانیابی با رویکرد برنامه ریزی ریاضی و برحسب انواع تابع هدف، به سه دسته تقسیم شده اند :

۱- هدفهای کششی[۳۸] : این هدفها اشاره به نزدیکی هر چه بیشتر محل استقرار کارخانه به مشتریان و کمتر کردن مسافت دارند که شامل قدیمی ترین مسائل مکانیابی می شوند. در واقع مسائلی که تابع هدف آنها به صورت کمینه سازی است، هدفهای کششی دارند.

۲- هدفهای فشاری[۳۹] : این هدفها مسائل مکانیابی مراکز نامطلوب را در بر می گیرند و از اوایل دهه ۱۹۷۰ بوجود آمدند. هدف در این مسائل، حداکثر کردن فاصله مراکز جدید از مراکز موجود است.

۳- هدفهای متعادل[۴۰] : هدفهایی هستند که تلاش در متعادل ساختن مسافت بین مراکز و مشتریان دارند. این هدفها پیوسته ترین نوع هدفها هستند و هدف اصلی آنها دستیابی به برابری است.

---

[36] Mini sum
[37] Minimax

[38] Pull purposes
[39] Push purposes
[40] Balancing purposes

۲-۱-۲ انواع مسایل مکانیابی

مسائل مکانیابی دارای تنوع بسیار زیادی هستند، از این رو برای سهولت در بیان، این مسائل را به راه های مختلفی دسته بندی کرده اند، مانند مسائل مکانیابی نقطه ای، مکانیابی پیوسته، یا مسائل مکانیابی p-میانه[۴۱] ، مکانیابی p-مرکز[۴۲] ، مسائل تخصیص[۴۳] ، مکانیابی پوششی[۴۴] وغیره. برخی از عناصر در دسته بندی مسایل مکانیابی نقش مهمی دارند. از این رو در حین دسته بندی مسائل مکانیابی باید عناصری مانند انواع مراکز جدید، مکان مراکز موجود، برهمکنش مراکز موجود و جدید، مشخصات فضای جواب، اندازه فاصله، تلفیق با سایر مسایل، تقاضا، ظرفیت، نوع مراکز، قطعی و احتمالی بودن داده ها، تواتر اجرا، تنوع محصول و تابع هدف مورد توجه قرار گیرند .

۲-۱-۳ اندازه گیری فاصله در مسائل مکانیابی

یک عنصر اساسی در فرمولبندی کردن مسائل مکانیابی، مفهوم فاصله است. کوتاه ترین فاصله بین دو نقطه خط راست است. اما در مدل های واقعی و عملی این فاصله به ندرت بدست می آید.

تعریف ۲-۱: فاصله ℓ_p بین نقاط $a_i=(a_{i1},a_{i2})$ و $a_j=(a_{j1},a_{j2})$ به صورت زیر است:

$$\ell_p(a_i,a_j)=\left(|a_{i1}-a_{j1}|^p+|a_{i2}-a_{j2}|^p\right)^{1/p} \qquad 1\le p<\infty$$

فاصله مستطیلی[۴۵] بین دو نقطه نمونه ای از متر با فاصله نسبتاً بلند می باشد. و حالت خاصی از نرم ℓ_p با p=1 می باشد. فاصله مستطیلی بین نقاط $a_i=(a_{i1},a_{i2})$ و $a_j=(a_{j1},a_{j2})$ به صورت زیر است :

$$\ell_1(a_i,a_j)=|a_{i1}-a_{j1}|+|a_{j2}-a_{j2}|$$

فاصله خط راست (اقلیدسی) [۴۶] در بین دو نقطه نمونه ای از متر با فاصله نسبتاً کوتاه است. و حالت خاصی از نرم ℓ_p با p=2 می باشد. فاصله خط راست بین نقاط $a_i=(a_{i1},a_{i2})$ و $a_j=(a_{j1},a_{j2})$ نیز به صورت زیر می باشد:

---

[41] Median
[42] p-center
[43] Allocation
[44] Location
[45] Rectangular distance
[46] Straight-line(Euclidean) distance

$$\ell_2(a_i, a_j) = \left((a_{i1} - a_{j1})^2 + (a_{i2} - a_{j2})^2\right)^{1/2}$$

۲-۲ مساله مکانیابی تک وسیله ای[۴۷]

مساله مکانیابی تک وسیله ای یکی از انواع مختلف مسائل مکانیابی است. در این مساله هدف یافتن نقطه یا مکان وسیله جدید[۴۸] (سرویس دهنده) در فضای مورد نظر و در بین نقاطی داده شده به گونه ای است که تابع هدف بهینه گردد. نقاطی (وسایل) که در صفحه موجودند و در حقیقت مکان مشتری را نشان می دهند، نقاط موجود[۴۹] نامیده می شوند. مفهوم فاصله بین نقطه جدید و نقاط موجود استفاده و کارکرد مترها را موجب خواهد شد.

مکان های نقطه جدید و نقاط موجود ممکن است در یک فضای گسسته مثل شبکه، یا پیوسته مثل فضاهای دو یا سه بعدی باشد، همچنین این مکان ها ممکن است زیر مجموعه ای از فضای دو یا سه بعدی نیز باشند. دو نوع از تابع هدف های این مساله در این پایان نامه مورد بررسی قرار خواهد گرفت عبارتند از: مساله مکانیابی تک وسیله ای با کمترین مجموع و دیگری مساله مکانیابی تک وسیله ای مینیماکس.

۲-۲-۱ مساله مکانیابی تک وسیله ای بر روی صفحه با تابع هدف کمترین مجموع

هدف این مساله پیدا کردن یک نقطه (مکان) جدید بر روی صفحه است به قسمی که مجموع فاصله آن از نقاط موجود داده شده روی صفحه کمینه گردد[۱۴]. یافتن مکان بهینه جدید معادل است با حل مساله بهینه سازی زیر:

$$Min \quad W(X) = \sum_{j=1}^{n} w_j \ell_p(X, a_j) \qquad (۱-۲)$$

که در آن

n: تعداد نقاط موجود در صفحه،

w_j: هزینه انتقال از نقطه جدید تا نقطه موجود a_j و ($w_j > 0$)،

$a_j = (a_{j1}, a_{j2})$: موقعیت نقطه موجود a_j روی صفحه،

$X = (x_1, x_2)$: موقعیت نقطه جدید روی صفحه،

---

[47] Location facilities

[48] New facility

[49] Existing facility

$\ell_p(X, a_j)$: فاصله بین نقطه جدید با نقطه موجود a_j .

۲-۲-۱-۱ بررسی مساله با فاصله خط راست

مساله (۲-۱) با تابع فاصله خط راست (p=2 در ℓ_p) را در نظر بگیرید، داریم:

$$Min \quad W(X) = \sum_{j=1}^{n} w_j \left((x_1 - a_{j1})^2 + (x_2 - a_{j2})^2 \right)^{1/2} \qquad (۲-۲)$$

این مساله را با مفهوم تابع محدب حل خواهیم کرد.

تعریف ۲-۴: تابع $f(x)$ را محدب[۵۰] گویند، هرگاه پاره خط واصل بین هر دو نقطه $(x^1, f(x^1))$ و $(x^2, f(x^2))$ روی گراف تابع، هرگز زیر نمودار قرار نگیرد. به عبارتی دیگر تابع $f(x)$ محدب است، اگر به ازای هر x^1 و x^2 که دو نقطه مجزا در دامنه f می باشند و هر $\lambda \in [0,1]$ داشته باشیم:

$$f(\lambda x^1 + (1-\lambda)x^2) \le \lambda f(x^1) + (1-\lambda) f(x^2)$$

تابع $f(x)$ را کیدا محدب[۵۱] گویند، هرگاه به ازای هر x^1 و x^2 که دو نقطه مجزا در دامنه f می باشند و هر $\lambda \in (0,1)$ داشته باشیم:

$$f(\lambda x^1 + (1-\lambda)x^2) < \lambda f(x^1) + (1-\lambda) f(x^2)$$

گزاره ۲-۱ ([۱۴]) : تابع $w_j \ell_2(X, a_j)$ تابعی محدب از X است.

اثبات: چون w_jها ثابت های مثبت هستند، می توان بدون از دست دادن کلیت مساله $w_j = 1$ قرار داد. همچنین a_{j1} و a_{j2} ها نیز اعداد ثابت هستند. بنابراین برای اثبات گزاره کافیست ثابت کنیم $f(y_1, y_2) = (y_1^2 + y_2^2)^{1/2}$ محدب می باشد. که در این رابطه فقط مبدا مختصات به (a_{j1}, a_{j2}) تغییر یافته است.

برای محدب بودن، کافی است به ازای $y' = (y_1', y_2')$ و $y'' = (y_1'', y_2'')$ نشان دهیم:

$$\left((\lambda y_1' + (1-\lambda) y_1'')^2 + (\lambda y_2' + (1-\lambda) y_2'')^2 \right)^{1/2}$$
$$\le \lambda \left((y_1')^2 + (y_2')^2 \right)^{1/2} + (1-\lambda) \left((y_1'')^2 + (y_2'')^2 \right)^{1/2}$$

---

[50] Convex function
[51] Strictly convex function

برای نشان دادن درستی این رابطه، از نامساوی مثلث بردارها استفاده می کنیم. برای دو بردار حقیقی $p=(p_1,p_2)$ و $q=(q_1,q_2)$ رابطه زیر برقرار است:

$$\left((p_1+q_1)^2+(p_2+q_2)^2\right)^{1/2}\le\left(p_1^2+p_2^2\right)^{1/2}+\left(q_1^2+q_2^2\right)^{1/2}$$

حال اگر قرار دهیم: $p_1=\lambda y_1'$ و $p_2=\lambda y_2'$ و $q_1=(1-\lambda)y_1''$ و $q_2=(1-\lambda)y_2''$ رابطه تحدب برقرار می شود.

تعریف ۲-۵: نقطه $x^*=(x_1^*,...,x_n^*)$ یک نقطه بهینه موضعی[۵۲] برای مساله مینیمم سازی (ماکزیمم سازی)، $\min(\max)f(x)$ است، هرگاه x^* یک نقطه مینیمم (ماکزیمم) موضعی برای مساله $\min(\max)f(x)$ باشد.

تعریف ۲-۶: نقطه x^* یک نقطه بهینه سراسری (مطلق)[۵۳] برای مساله مینیمم سازی (ماکزیمم سازی)، $\min(\max)f(x)$ است، هرگاه x^* یک نقطه مینیمم (ماکزیمم) سراسری برای مساله $\min(\max)f(x)$ باشد.

توجه کنید که گزاره (۲-۱) زمانی برقرار است که وزن ها مثبت باشند. می توان نشان داد که مجموع توابع محدب، محدب است[۱۴]. ازاین رو $W(X)$ نیز محدب می باشد و این بدین معنی است که در مساله (۲-۲) بهینه موضعی، بهینه سراسری می باشد و همچنین $W(X)$ نقطه عطف ندارد. با این اطلاعات مطمئن می شویم که معادلات اکسترمال[۵۴] $W(X)$ که به صورت زیر هستند می توانند بهینه سراسری را برای مساله (۲-۲) تولید کنند.

$$\frac{\partial W(X)}{\partial x_k}=\sum_{j=1}^{n}\frac{w_j(x_k-a_{jk})}{\ell_2(X,a_j)}=0 \qquad k=1,2 \qquad (۳-۲)$$

اما مشکلی که بوجود می آید این است که اگر $\ell_2(X,a_j)=0$ ، معادلات مشتق (۲-۳) در آن تعریف نشده است. بنابراین اگر مکان بهینه برای نقطه جدید بر روی یکی از نقاط موجود منطبق گردد، آنگاه معادله (۲-۳) نمی تواند برای بدست آوردن جواب بهینه مورد استفاده قرار گیرد. بنابراین به راحتی می توان هر یک از نقاط موجود را جداگانه برای بهینگی بررسی کرد.

---

[52] Local optima
[53] Global optima
[54] Extremal equations

گزاره ۲-۲[مینیمم $W(X)$ در یکی از نقاط موجود مکانیابی] ([۱۴]): $W(X)$ در r امین نقطه موجود مینیمم می شود اگروفقط اگر

$$CR_r = \left[\left(\sum_{\substack{j=1 \\ \neq r}}^{n} \frac{w_j (a_{r1} - a_{j1})}{\ell_2(a_1, a_j)} \right)^2 + \left(\sum_{\substack{j=1 \\ \neq r}}^{n} \frac{w_j (a_{r2} - a_{j2})}{\ell_2(a_r, a_j)} \right)^2 \right]^{1/2} \leq w_r \qquad (۴-۲)$$

اثبات: یک جابجایی از مکان جدید (x_1, x_2) با فاصله t در جهت بردار $d = (d_1, d_2)$ یعنی $(x_1 + td_1, x_2 + td_2)$ را در نظر می گیریم، که در آن $(d_1^2 + d_2^2)^{1/2} = 1$ می باشد. حال نرخ تغییر $W(X + td)$ را نسبت به t، وقتی $X = a_r$ و t به صفر میل می کند را می یابیم:

$$\begin{aligned}
\frac{dW}{dt} &= \sum_{j=1}^{n} \frac{w_j \left((a_{r1} + td_1 - a_{j1}) d_1 + (a_{r2} + td_2 - a_{j2}) d_2 \right)}{\left((a_{r1} + td_1 - a_{j1})^2 + (a_{r2} + td_2 - a_{j2})^2 \right)^{1/2}} \\
&= \frac{w_r (td_1^2 + td_2^2)}{\left((td_1)^2 + (td_2)^2 \right)^{1/2}} \\
&+ d_1 \sum_{\substack{j=1 \\ \neq r}}^{n} \frac{w_j (a_{r1} + td_1 - a_{j1})}{\left((a_{r1} + td_1 - a_{j1})^2 + (a_{r2} + td_2 - a_{j2})^2 \right)^{1/2}} \\
&+ d_2 \sum_{\substack{j=1 \\ \neq r}}^{n} \frac{w_j (a_{r2} + td_2 - a_{j2})}{\left((a_{r1} + td_1 - a_{j1})^2 + (a_{r2} + td_2 - a_{j2})^2 \right)^{1/2}}
\end{aligned}$$

که این مشتقات را می توان به صورت زیر نوشت:

$$\frac{dW}{dt} = w_r (d_1^2 + d_2^2)^{1/2} + d_1 R_1(t) + d_2 R_2(t)$$

که $R_1(t)$ و $R_2(t)$ در رابطه بالا تعریف شده بودند. بنابراین:

$$\frac{dW}{dt}\Big|_{t \to o} = w_r (d_1^2 + d_2^2)^{1/2} + d_1 R_1 + d_2 R_2 = w_r + d_1 R_1 + d_2 R_2$$

که

$$R_1 = \sum_{\substack{j=1 \\ \neq r}}^{n} \frac{w_r (a_{r1} - a_{j1})}{\ell_2(a_r, a_j)}$$

و

$$R_2 = \sum_{\substack{j=1 \\ \neq r}}^{n} \frac{w_r\left(a_{r2} - a_{j2}\right)}{\ell_2\left(a_r, a_j\right)}$$

با استفاده از محاسبات اولیه و شرط $d_1^2 + d_2^2 = 1$ می توان گفت که کمترین مقدار $\frac{dW}{dt}|_{t\to 0}$ به ازای:

$$d_1 = -\frac{R_1}{\left(R_1^2 + R_2^2\right)^{1/2}}\ ,\ d_2 = -\frac{R_2}{\left(R_1^2 + R_2^2\right)^{1/2}}$$

اتفاق می افتد و بنابراین:

$$\min\ \frac{dW}{dt}|_{t\to 0} = w_r - \left(R_1^2 + R_2^2\right)^{1/2}$$

وقتی که مشتقات مثبت هستند، هر بار که $\left(x_1, x_2\right)$ در هر جهت $\left(a_{r1}, a_{r2}\right)$ حرکت می کند، $W(X)$ افزایش می یابد. چون $W(X)$ محدب است، $X^* = \left(a_{r1}, a_{r2}\right)$ نقطه مینیمم است اگر و فقط اگر $w_r \geq \left(R_1^2 + R_2^2\right)^{1/2} = CR_r$.

روش های تکراری مختلفی برای یافتن جواب مساله (۲-۱) وجود دارد. یکی از قدیمی ترین و شاید ساده ترین آنها روش وایس فیلد[۵۵] است.

روش وایس فیلد برای یافتن جواب مساله (۲-۲): برای بدست آوردن کمینه W (X)، فرض کنید موقتاً امکان تطابق نقطه جدید با یکی از نقاط موجود وجود نداشته باشد.
روش تکراری برای مکانیابی بهینه با بازنویسی معادله (۲-۳) به صورت زیر بدست می آید.

$$x_k = \frac{\sum_{j=1}^{n} \frac{w_j a_{jk}}{\ell_2\left(X, a_j\right)}}{\sum_{j=1}^{n} \frac{w_j}{\ell_2\left(X, a_j\right)}} \qquad k = 1,2 \qquad (۵-۲)$$

توجه کنید که x_k که در طرف چپ معادله (۲-۵) است، واقعاً ایزوله (تنها) نیست، زیرا هر $\ell_2\left(X, a_j\right)$ تابعی از x_k می باشد، هر چند که معادله (۲-۵) را می توان برای میل (نزدیکی) به $\left(x_1^*, x_2^*\right)$ که مکان بهینه نقطه جدید است، استفاده کرد.

---

[55] Weiszfeld procedure

فرض کنید ℓ تکرار کامل داریم و مکان $\left(x_1^{(\ell)}, x_2^{(\ell)}\right)$ را به دست آورده ایم، آنگاه با استفاده از معادله (۲-۵) برای $(\ell+1)$امین تکرار داریم:

$$x_k^{(\ell+1)} = \frac{\sum_{j=1}^{n} \frac{w_j a_{jk}}{\ell_2\left(X^{(\ell)}, a_j\right)}}{\sum_{j=1}^{n} \frac{w_j}{\ell_2\left(X^{(\ell)}, a_j\right)}} \qquad k = 1,2 \qquad (۲-۶)$$

قبل از شروع تکرارها به یک نقطه آغازین $\left(x_1^{(0)}, x_2^{(0)}\right)$ نیازمندیم. یک راه انتخاب نقطه اولیه، انتخاب جواب مساله فاصله توان دو اقلیدسی مساله (۲-۱) است. که مشابه مساله (۲-۲) است با این تفاوت که فاصله $\ell_2\left(X, a_j\right)$، به توان دو می باشد. جواب بهینه مساله فاصله توان دو اقلیدسی به صورت زیر است [۱۴]:

$$\min W(X) = \sum_{j=1}^{n} w_j \left[\left(x_1 - a_{j1}\right)^2 + \left(x_2 - a_{j2}\right)^2\right]$$

$$x_k = \frac{\sum_{j=1}^{n} w_j a_{jk}}{\sum_{j=1}^{n} w_j} \qquad k = 1,2 \qquad (۲-۷)$$

ابن نقطه مرکز گرانش نقاط موجود نام دارد و می تواند به عنوان نقطه شروع روش تکراری (۲-۶) استفاده شود.

در این رویه، تکرارها به یک مکان بهینه میل خواهد کرد. فرض بر این است که نه مکان بهینه و نه هیچ یک از تکرارها بر مکان های موجود منطبق نیستند. اگر چه مشکلات همگرایی را زمانی که $\left(x_1^*, x_2^*\right)$ به یکی از نقاط موجود منطبق می شود انتظار خواهیم داشت، اما عموماً این مشکلات بوجود نمی آیند.

هنگامی که CR_r در شرط (۲-۴) خیلی کوچکتر از w_r باشد، همگرایی نسبتاً سریع است و زمانی که $\ell_2\left(X, a_r\right)$ در معادله (۲-۵) تقریباً برابر صفر شود، مشکلات محاسبات می تواند رخ دهد. تجربه نشان می دهد که وقتی شرط (۲-۴) نزدیک به تساوی باشد، همگرایی ممکن است کند باشد. بنابراین بهتر است ابتدا تمام نقاط موجود مکانیابی را با استفاده از شرط (۲-۴) بررسی کنیم، اگر شرط (۲-۴) برقرار نباشد اما CR_r در بعضی نقاط موجود r برابر با w_r گردد، آنگاه تکرارها می توانند از نزدیک آن نقطه شروع شوند.

خاتمه این روش، معیار توقف آن، یعنی کران پایین مقدار بهینه $W(X)$ می باشد. این کران پایین دائماً در طی تکرارها به روز[۵۶] می شود. برای بدست آوردن این کران نیازمند گزاره زیر هستیم که اثبات آن در [۱۴] آمده است.

---

[56] Update

گزاره ۲-۳[دامنه پوسته محدب] ([۱۴]): جواب بهینه مساله (۲-۱) باید درون Ω، پوسته محدب نقاط موجود مکانیابی، قرار گیرد.

می توان نشان دادکه نقطه شروع تعریف شده با معادله (۲-۷) در این پوسته قرار دارد. بعلاوه به غیر از نقطه شروع استفاده شده در رویه (۲-۶)، نقاط بعدی نیز در پوسته محدب خواهد بود. حال یک منطق هندسی برای روش وایس فیلد ارائه می گردد.

برای شروع یادآوری می کنیم که $W(X)$ تابعی محدب است. این بدان معنی است که صفحه مماس گراف محدب کاسه شکل $W(X)$، در نقطه داده شده $X^{(\ell)}$، $W(X)$ را به ازای هر X تخمین می زند. بخصوص اگر مشتقات جزئی[۵۷] داده شده در معادله (۲-۵) موجود باشند. این یعنی به ازای هرX ;

$$W(X) \geq W\left(X^{(\ell)}\right) + \sum_{K=1}^{2}\left[\frac{\partial W\left(X^{(\ell)}\right)}{\partial x_k}\right]\left(x_k - x_k^{(\ell)}\right)$$

گرادیان$\nabla W\left(X^{(\ell)}\right)$ برداری با مولفه های بدست آمده، به ترتیب مشتقات جزئی است. با انتخاب $X = X^*$ می توان نوشت:

$$\begin{aligned} W\left(X^*\right) &\geq W\left(X^{(\ell)}\right) + \nabla W\left(X^{(\ell)}\right)\cdot\left(X^* - X^{(\ell)}\right) \\ &\geq W\left(X^{(\ell)}\right) - \left|\nabla W\left(X^{(\ell)}\right)\cdot\left(X^* - X^{(\ell)}\right)\right| \\ &\geq W\left(X^{(\ell)}\right) - \left\|\nabla W\left(X^{(\ell)}\right)\right\| \left\|X^* - X^{(\ell)}\right\| \end{aligned}$$

(چون طبق نامساوی شوارتز[۵۸] داریم: $|u.v| \leq \|u\| \|v\|$)

(. نشان دهنده ضرب برداری و $\|.\|$ قدر یک بردار است)

به ازای $\ell \geq 1$ ، X^* و $X^{(\ell)}$ هر دو در پوسته محدب Ω هستند. از این رو $\left\|X^* - X^{(\ell)}\right\|$ نمی تواند بزرگتر از فاصله خط راست $\sigma\left(X^{(\ell)}\right)$ (فاصله خط راست بین نقطه $X^{(\ell)}$ و دورترین نقطه از $X^{(\ell)}$ در Ω) گردد. بنابراین کران بالای رو به رشد در$W(X)$ باید $\left\|\nabla W\left(X^{(\ell)}\right)\right\| \sigma\left(X^{(\ell)}\right)$ باشد.

گزاره ۲-۴ [کران پایین $W\left(X^*\right)$] ([۱۴]):

$$W\left(X^*\right) \geq LB^{(\ell)} = W\left(X^{(\ell)}\right) - \left\|\nabla W\left(X^{(\ell)}\right)\right\| \sigma\left(X^{(\ell)}\right) \qquad (۲-۸)$$

که در آن $\sigma\left(X^{(\ell)}\right) = \max_{y\in\Omega}\{\ell_2\left(X^{(\ell)}, y\right)\}$ می باشد.

---

[57] Partial derivatives

[58] Schwartz

بنابراین دانستن کران بالا یا کران پایین جهت بهبود مقدار تابع هدف در هر تکرار روش وایس فیلد، میسر است. مبنای معیار توقف روی زیر بهینگی متناسب[۵۹] با مجموعه زیر است:

$$S^{(\ell)} = \frac{\left\|\nabla W\left(X^{(\ell)}\right)\right\| \sigma\left(X^{(\ell)}\right)}{LB^{(\ell)}} \qquad (۹-۲)$$

از نامساوی (۲-۸) هنگامی که $LB^{(\ell)} > 0$ داریم:

$$S^{(\ell)} \geq \frac{\left(W\left(X^{(\ell)}\right) - W\left(X^{*}\right)\right)}{W\left(X^{*}\right)}$$

از این رو اگر می خواهیم یک تکرار $X^{(\ell)}$ را با زیربهینگی نسبی کراندار شده به صورت $\frac{W\left(X^{(\ell)}\right) - W\left(X^{*}\right)}{W\left(X^{*}\right)} < \varepsilon$ پیدا کنیم، تکرارها باید تا $S^{(\ell)} < \varepsilon$ ادامه یابد. به عنوان مثال، اگر تکرارها زمانی که $S^{(\ell)} < 0/001$ است پایان یابند، آنگاه $W\left(X^{(\ell)}\right)$، ۰/۱٪مقدار مینیمم $W\left(X^{*}\right)$ می باشد.

مثال۲-۱ ([۱۴]): چهار نقطه (۱،۱), (۱،۴) , (۲،۲), (۴،۵) به عنوان نقاط موجود, به ترتیب با وزن های ۱و۲و۲و۴ داده شده است. با استفاده از فاصله خط راست مکان بهینه ای برای نقطه جدید بیابید.

جواب: با توجه به مقادیر CR_r در جدول ۲-۱ ([۱۴])، چون مقدار CR_r ها از w_rها بزرگتر است، بنابراین نقطه بهینه جدید بر هیچ یک از نقاط مکانیابی موجود منطبق نمی شود. جدول ۲-۲ تکرارها را با شروع از مرکز گرانشی نقاط نشان می دهد.

جدول ۲-۱ محاسبه CR_r

مختصات نقاط موجود مکانیابی	r	CR_r	w_r
(1,1)	1	7.635	1
(1,4)	2	4.931	2
(2,2)	3	4.435	2
(4,5)	4	4.754	4

[59] Proportional suboptimality

جدول۲-۲ تکرارهای مثال۲-۱

ℓ	$S^{(\ell)}$	$LB^{(\ell)}$	مقدار تابع هدف	مختصات نقطه جدید
	7.55×10^{-2}	16.407	17.646	(2.556,3.667)
	2.41×10^{-2}	17.210	17.624	(2.523,3.745)
	1.376×10^{-2}	17.382	17.621	(2.527,3.772)
	1.024×10^{-2}	17.441	17.620	(2.536,3.785)
	7.934×10^{-3}	17.481	17.619	(2.544,3.794)
	6.192×10^{-3}	17.511	17.619	(2.551,3.799)
	4.847×10^{-3}	17.534	17.619	(2.557,3.804)
	3.802×10^{-3}	17.552	17.619	(2.560,3.807)
	2.987×10^{-3}	17.566	17.619	(2.564,3.810)
	2.350×10^{-3}	17.577	17.619	(2.567,3.812)
	1.851×10^{-3}	17.586	17.619	(2.569,3.814)
	5.660×10^{-4}	17.608	17.618	(2.576,3.818)
	1.742×10^{-4}	17.615	17.618	(2.577,3.820)
	1.580×10^{-4}	17.618	17.618	(2.577,3.820)

۲-۲-۱-۲ بررسی مساله با فاصله مستطیلی

فاصله مستطیلی در اغلب مسائل مکانیابی کاربرد دارد و همچنین به خاطر سادگی اغلب مسائلی که با فاصله خط راست نسبتاً سخت هستند از این فاصله استفاده می کنند.
بنابراین برای $p=1$ ، مساله (۲-۱) به صورت زیر تبدیل می شود:

$$\min W(X)=\sum_{j=1}^{n} w_j\left(\left|x_1-a_{j1}\right|+\left|x_2-a_{j2}\right|\right) \qquad (۱۰-۲)$$

دو خاصیت مفید برای یافتن (x_1^*,x_2^*) در این حالت وجود دارد. اولین آن تفکیک پذیری است. بنابراین مساله (۲-۱۰) را می توان به صورت زیر نوشت:

$$\min_X W(X) = W_1(X) + W_2(X)$$

s.t

$$W_1(x_1) = \sum_{j=1}^{n} w_j |x_1 - a_{j1}| \qquad (۱۱-۲)$$

$$W_2(x_2) = \sum_{j=1}^{n} w_j |x_2 - a_{j2}|$$

کمینه کردن $W(X)$ معادل است با یافتن جداگانه x_1 برای مینیمم کردن $W_1(x_1)$ و x_2 برای کمینه ساختن $W_2(x_2)$. بنابراین مساله به صورت زیر تبدیل می شود:

$$\min W_k(x_k) = \sum_{j=1}^{n} |x_k - a_{jk}| \qquad k = 1,2 \qquad (۱۲-۲)$$

که این مساله خیلی راحت تر قابل حل می باشد.

برای حل مساله، ابتدا باید بررسی کنیم که $w_j |x_k - a_{jk}|$ یک تابع محدب از x_k است. با ترسیم ساده تابع برحسب x_k به ازای هر w_j و a_{jk} می توان نشان داد که این تابع محدب است و چون مجموع توابع محدب، محدب است بنابراین $W_k(x_k)$ نیز محدب می باشد.

برای تحلیل آسان تر حل، علامت گذاری (۲-۱۲) را تغییر می دهیم و مقادیر a_{jk} را به ازای هر j دوباره مرتب می کنیم، به صورتی که $a_{(1)k} < a_{(2)k} < a_{(3)k} < \ldots < a_{(n_k)k}$ باشد و فرض می کنیم که $w_1^k, w_2^k, \ldots, w_{n_k}^k$ وزن های متناظر باشند. توجه کنید که اکنون تعداد n_k مختصات وجود خواهد داشت که $n_k < n$.

مثلاً فرض کنید اگر $a_{11} = a_{21} = 4$ و $w_1 = 3$ و $w_2 = 2$ باشد، آنگاه می توان مجموع جملات $3|x_1 - 4| + 2|x_1 - 4|$ را به یک جمله $5|x_1 - 4|$ تبدیل کرد.

بنابراین می توان دنباله ای از $a_{(j)k}$ ها را ساخت که برحسب مقدار، اکیداً نزولی هستند. و در نتیجه ممکن است تعداد $a_{(j)k}$ ها کمتر از تعداد a_{jk} ها گردند. اندیس بالای k در w_j^k نشان دهنده ترتیب و ترکیب مطابق وزن هاست. به همین دلیل است که ترتیب وزن ها در بعد x_1 با بعد x_2 متفاوت است و همچنین اندیس بالای k را برای تشخیص این ترتیب ها استفاده می کنیم.

مساله (۲-۱۲) به صورت زیر نوشته می شود:

$$\min W_k(x_k) = \sum_{j=1}^{n} w_j^k |x_k - a_{(j)k}| \qquad k = 1,2 \qquad (۱۳-۲)$$

حال تابع $W_k(x_k)$ فرمی مناسب برای محاسبه مشتق دارد و می توان نوشت:

$$W_K'(x_k) = -\sum_{j=1}^{n_k} w_j^k \qquad x_k < a_{(1)k} \qquad (۱۴\ a-۲)$$

$$W_k'(x_k) = \sum_{j=1}^{t} w_j^k - \sum_{j=t+1}^{n_k} w_j^k \qquad a_{(t)k} < x_k < a_{(t+1)k} \qquad (۱۴\ b-۲)$$

$$W_k'(x_k)=\sum_{j=1}^{n_k} w_j^k \qquad x_k > a_{(n_k)k} \qquad (۱۴\ \text{c-}۲)$$

مشاهده می کنیم که شیب تابع $W_k(x_k)$ از قطعات خطی ساخته شده است و تنها در نقاط $a_{(j)k}$ تغییر می کند. در مجموع می توان گفت $W_k(x_k)$ تابعی پیوسته-محدب و تکه ای-خطی است که مشتق اول آن در نقاط a_{jk} ناپیوسته است. چون شیب $W_k(x_k)$ در معادله (۲-۱۴) با افزایش t افزایش می یابد، بنابراین مینیمم $W_k(x_k)$ هر جا که شیب تابع از منفی به مثبت یا در بعضی نقاط از منفی به صفر تغییر می کند اتفاق می افتد. و در حالتی دیگر مینیمم تابع روی یک فاصله از x_k اتفاق می افتد، هرچند که حداقل یک نقطه $a_{(t)k}$ باید مینیمم کننده $W_k(x_k)$ باشد.

گزاره ۲-۵ [شرایط برای مینیمم $W_k(x_k)$] ([۱۴]): فرض کنید x^* جواب بهینه مساله (۲-۱۳) باشد و

$$\sum_{j=1}^{t-1} w_j^k - \sum_{j=t}^{n_k} w_j^k < 0 \qquad (۱۵\ \text{a-}۲)$$

و

$$\sum_{j=1}^{t} w_j^k - \sum_{j=t+1}^{n_k} w_j^k \geq 0 \qquad (۱۵\ \text{b-}۲)$$

در نقطه ای مانند t^* برقرار باشد،

اگر شرط (۲-۱۵ b) نامساوی اکید باشد، آنگاه $x_k^* = a_{(t^*)k}$، یعنی $a_{(t^*)k}$ جواب بهینه است

و اگر شرط(۲-۱۵ b) نامساوی باشد، آنگاه $x_k^* \in \lfloor a_{(t^*)k}, a_{(t^*+1)k} \rfloor$ می شود.

حال شرایط گزاره (۲-۱۵) را می توان در قالبی مناسب تر برای یافتن t^* بیان کرد. بنابراین شرط (۲-۱۵ a) را می توان به صورت زیر نوشت:

$$\sum_{j=1}^{t-1} w_j^k + \sum_{j=1}^{t-1} w_j^k - \sum_{j=1}^{n_k} w_j^k < 0$$

و شرط (۲-۱۵ b) را به صورت زیر نوشت:

$$\sum_{j=1}^{t} w_j^k + \sum_{j=1}^{t} w_j^k - \sum_{j=1}^{t} w_j^k - \sum_{j=t+1}^{n_k} w_j^k \geq 0$$

اگر قرار دهیم:

$$C=\sum_{j=1}^{n_k} w_j^k \qquad k=1,2$$

آنگاه شرایط گزاره (۲-۱۵) به صورت زیر می شود:

$$-C+2\sum_{j=1}^{t-1} w_j^k < 0 \qquad (۱۶\ \text{a-}۲)$$

$$-C+2\sum_{j=1}^{t} w_j^k \geq 0 \qquad (۱۶\ \text{b-}۲)$$

حال نامساوی های (۲-۱۶) یک روش محاسباتی برای حل مساله پیشنهاد می کند. از $t=1$ شروع می کنیم و دو برابر مجموع وزن های نقاط $a_{(t)k}$ را تا اولین نقطه ای که با C برابر شود یا از آن بیشتر گردد محاسبه می کنیم. حال نقطه بهینه x_k^* همان اولین نقطه می باشد.

مثال۲-۲: چهار نقطه (۱,۱) ، (۲,۴) ، (۲,۳) و (۴,۲) را به ترتیب با وزن های ۲ ، ۳ ، ۱ و ۲ در نظر بگیرید و با استفاده از فاصله مستطیلی مکان بهینه ای برای نقطه جدید بدست آورید.

حل:

جدول ۲-۳ جهت حل مثال(۲-۲)

ترتیب در بعد x_2		ترتیب در بعد x_1		مختصات و وزن های اصلی			
w_j^2	$a_{(j)2}$	w_j^1	$a_{(j)1}$	w_j	a_{j2}	a_{j1}	j
۲	۱	۲	۱	۲	۱	۱	۱
۲	۲	۴	۲	۳	۴	۲	۲
۱	۳	۲	۴	۱	۳	۲	۳
۳	۴			۲	۲	۴	۴

با استفاده از این جدول محاسبات زیر را انجام می دهیم:

$$C=\sum_{J=1}^{n_k} w_j^1 = ۲+۴+۲=۸$$

حال برای یافتن x_1^* محاسبات زیر را داریم:

$$-C+2\sum_{j=1}^{1} w_j^1 = -۸+۴=-۴$$

$$-C+2\sum_{j=1}^{2} w_j^1 = -۸+۱۲=۴>۰$$

چون شرط (۲-۱۶ b) به صورت اکید برقرار است، بنابراین $x_1^* = a_{(2)1} = 2$

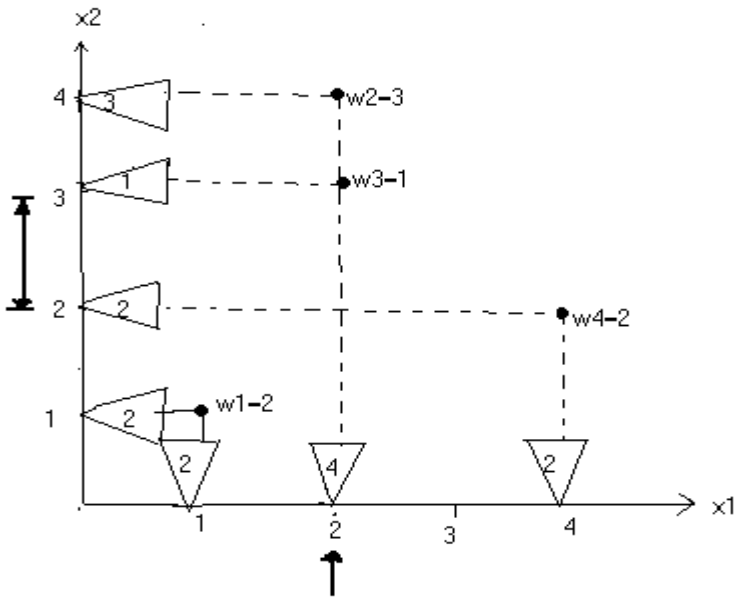
و برای یافتن x_2^* نیز داریم:

$$-C+2\sum_{j=1}^{1} w_j^1 = -۸+۴=-۴$$

$$-C+2\sum_{j=1}^{2} w_j^1 = -۸+۸=۰$$

چون شرط (۲-b ۱۶) به صورت نامساوی برقرار است، بنابراین $x_2^* \in [a_{(2)2}, a_{(3)2}] = [2,3]$.

اما روش راحت تری برای حل این مساله با استفاده از نمودار به صورت زیر وجود دارد:
وزن ها را بر روی تصویر نقاط در محور x_1 و x_2 می نویسیم (شکل۲-۱). محور x_1 را در آنجایی که وزن ها نصف می شود به دو قسمت تقسیم می کنیم. که در مثال (۲-۲) این نقطه $x_1 = 2$ می باشد. به همین صورت وزن ها در مثال (۲-۲) روی محور x_2 زمانی نصف می شود که $2 \le x_2 \le 3$ باشد. بنابراین مقدار جواب بهینه به دست می آید که این جواب ها در شرایط (۲-۱۵) و (۲-۱۶) صادق هستند.



شکل ۲-۱ نمایش وزن میانی

۲-۲-۱-۳ بررسی مساله با فاصله ℓ_p

مساله (۲-۱) با فاصله ℓ_p به صورت زیر است:

$$\min W(X)=\sum_{j=1}^{n} w_j \left(\left|x_1-a_{j1}\right|^p+\left|x_2-a_{j2}\right|^p\right)^{1/p} \qquad (۱۷-۲)$$

ابتدا دو خاصیتی که $\ell_p(X,a_j)$ را مشخص می کند بیان می کنیم:

۱- مقدار $\ell_p(X,a_j)$ باافزایش p ، کاهش می یابد. یعنی برای $X\neq a_j$ و $p<p'$ داریم:

$$\left(\left|x_1-a_{j1}\right|^p+\left|x_2-a_{j2}\right|^p\right)^{1/p} > \left(\left|x_1-a_{j1}\right|^{p'}+\left|x_2-a_{j2}\right|^{p'}\right)^{1/p'}$$

۲- هرگاه $p\to\infty$ ، آنگاه $\ell_p(X,a_j)$ بزرگتر از $\left|x_1-a_{j1}\right|$ و $\left|x_2-a_{j2}\right|$ است.

نامساوی مینکوسکی: نامساوی مینکوسکی[۶۰] به صورت زیر است:

$$\left(\sum_{k=1}^{K}|\alpha_k+\beta_k|^p\right)^{1/p} \le \left(\sum_{k=1}^{K}|\alpha_k|^p\right)^{1/p}+\left(\sum_{k=1}^{K}|\beta_k|^p\right)^{1/p}$$

که $p\ge 1$ است و α_k و β_k اعداد حقیقی هستند.

گزاره ۲-۶ [تحدب $w_j\ell_p(X,a_j)$ **][۱۳]:** $w_j\ell_p(X,a_j)$ تابعی محدب از X می باشد.

اثبات: پیرو گزاره (۲-۱) ثابت می کنیم که $f(y_1,y_2)=\left(|y_1|^p+|y_2|^p\right)^{1/p}$ محدب است. با استفاده از نامساوی مینکوسکی خواهیم داشت:

$$\begin{aligned}
&f\left(\lambda y_1'+(1-\lambda)y_1'',\lambda y_2'+(1-\lambda)y_2''\right)\\
&\le \left(\left(|\lambda y_1'|+|(1-\lambda)y_1''|\right)^p+\left(|\lambda y_2'|+|(1-\lambda)y_2''|\right)^p\right)^{1/p} \qquad (*)\\
&\le \left(|\lambda y_1'|^p+|\lambda y_2'|^p\right)^{1/p}+\left(|(1-\lambda)y_1''|^p+|(1-\lambda)y_2''|^p\right)^{1/p} \qquad (**)\\
&=\lambda\left(|y_1'|^p+|y_2'|^p\right)^{1/p}+(1-\lambda)\left(|y_1''|^p+|y_2''|^p\right)^{1/p}\\
&=\lambda f(y_1',y_2')+(1-\lambda)f(y_1'',y_2'')
\end{aligned}$$

((*) : نامساوی مثلث و (**): نامساوی مینکوسکی می باشند)

---

[60] Minkowski

در نتیجه حکم ثابت شد.

هر جمله $W(X)$ در (۲-۱۷) محدب است، دوباره با استفاده از این حقیقت کـه مجمـوع توابـع محدب، محدب است، نتیجه می گیریم که $W(X)$ تابع محـدبی از X مـی باشـد. از ایـن رو یـک مینیمم موضعی، مینیمم سراسری می باشد.

گزاره ۲-۷[مینیمم $W(X)$ در نقاط موجود مکانیـابی][۱۴]: $W(X)$ در (a_{r1}, a_{r2}) دارای مقـدار مینیمم است، اگر و فقط اگر;

$$CRP_r = \left(|R_{r1}|^{p/(p-1)} + |R_{r2}|^{p/(p-1)}\right)^{(p-1)/p} \le w_r \qquad p>1 \qquad (۱۸\text{ a-}۲)$$

$$\max\left(|R_{r1}|, |R_{r2}|\right) \le w_r \qquad p=1 \qquad (۱۸\text{ b-}۲)$$

که

$$R_{rk} = \sum_{\substack{j=1 \\ \neq r}}^{n} \frac{w_j \, sign(a_{rk} - a_{jk}) |a_{rk} - a_{jk}|^{p-1}}{\left(\ell_p(a, a_j)\right)^{p-1}} \qquad k = 1,2$$

مشاهده می شود که در (۲-a ۱۸) داریم:

$$R_{rk} = \frac{\partial}{\partial x_k} \sum_{\substack{j=1 \\ \neq r}}^{n} w_j \ell_p(X, a_j)\Big|_{x_k = a_{rk}}$$

جملات سمت چپ نامساوی (۲-a ۱۸) در $p=1$ تعریف نشده است.
بـا اسـتفاده از خاصـیت (۲) فاصـله هـای ℓ_p و بـا فـرض کـاهش p بـه سـمت ۱، آنگـاه داریـم: $p' = \frac{p}{p-1} \to \infty$ و به راحتی نامساوی (۲-b ۱۸) را نتیجه خواهیم گرفت.

هر جمله $|y|$ در مساله (۲-۱۷) را با $(y^2+\varepsilon)^{1/2}$ که درآن ε یک عـدد مثبـت کوچـک اسـت، تعویض می کنیم. این تقریب همیشه بزرگتر از مقدار اصلی می باشد، اما هنگامیکـه $\varepsilon \to 0$ میـل می کند، این تقریب نیز به جمله اصلی میل خواهد کرد و مساله (۲-۱۷) با رابطـه زیـر تقریـب زده می شود:

$$\min_X \; WH(X) = \sum_{j=1}^{n} w_j \left[\left((x_1 - a_{j1})^2 + \varepsilon\right)^{p/2} + \left((x_2 - a_{j2})^2 + \varepsilon\right)^{p/2}\right]^{1/p} \qquad (۱۹\text{-}۲)$$

ملاحظه می کنیم که $WH(X)$ اکیداً محدب است و تمام مشتقات آن در هر نقطه ای پیوسـته است. سوالی که پیش می آید این است که اگر نقطه ای که $WH(X)$ را مینیمم می کنـد یـافتیم،

آیا این نقطه $W(X)$ را نیز مینیمم می کند؟ نشان داده می شود که به راحتی مـی تـوان مینـیمم کننده $W(X)$ را با مینیمم کننده $WH(X)$ با انتخاب مقادیر کوچک ε تقریب زد.

گزاره ۲-۸ [ماکزیمم اختلاف بین $WH(X)$ و $W(X)$] ([۱۴]):

$$\max\{WH(X)-W(X)\}\le\Delta(\varepsilon)=2^{1/p}\varepsilon^{1/2}\left(\sum_{j=1}^{n}w_j\right) \qquad (۲۰-۳)$$

اثبات: کافیست نشان دهیم که:

$$L_p(X,a_j)-\ell_p(X,a_j)\le 2^{1/p}\varepsilon^{1/2}$$

که در آن، $L_p(X,a_j)=\left(\left((x_1-a_{j1})^2+\varepsilon\right)^{p/2}+\left((x_2-a_{j2})^2+\varepsilon\right)^{p/2}\right)^{1/p}$ است.

قرار دهید: $y_1=|x_1-a_{j1}|\ge 0$ و $y_2=|x_2-a_{j2}|\ge 0$ و $y_3=\varepsilon^{1/2}>0$. آنگاه:

$$\begin{aligned}
L_p(X,a_j)&=\left[\left(y_1^2+y_3^2\right)^{p/2}+\left(y_2^2+y_3^2\right)^{p/2}\right]^{1/p}\\
&\le\left[\left((y_1+y_3)^2\right)^{p/2}+\left((y_2+y_3)^2\right)^{p/2}\right]^{1/p}\\
&=\left[|y_1+y_3|^p+|y_2+y_3|^p\right]^{1/p}\\
&\le\left[|y_1|^p+|y_2|^p\right]^{1/p}+\left[|y_3|^p+|y_3|^p\right]^{1/p}\\
&=\left[|x_1-a_{j1}|^p+|x_2-a_{j2}|^p\right]^{1/p}+\left[\varepsilon^{p/2}+\varepsilon^{p/2}\right]^{1/p}\\
&=\ell_p(X,a_j)+2^{1/p}\varepsilon^{1/2}
\end{aligned}$$

در نتیجه حکم اثبات می شود.

بنابراین اختلاف بین $W(X)$ و $WH(X)$ هرگز فراتر از $\Delta(\varepsilon)$ نمی رود.

جواب مساله (۲-۱۹)، جوابی برای مساله (۲-۱۷) با حداکثر $\Delta(\varepsilon)$ بیشتر از مقدار جواب بهینـه اش خواهد بود. برای مشاهده این مطلب فرض کنید X^* مینـیمم کننـده $W(X)$ و X^{**} مینـیمم کننـده $WH(X)$ باشـد. چـون $WH(X^*)-W(X^*)<\Delta(\varepsilon)$ و $WH(X^{**})<WH(X^*)$ ، داریـم: $WH(X^{**})-W(X^*)<\Delta(\varepsilon)$. بنابراین $W(X^{**})-W(X^*)<\Delta(\varepsilon)$.

روش وایس فیلد نیز می تواند برای حل مساله (۳-۱۷) مـورد اسـتفاده قـرار گیـرد. بـا قـرار دادن مشتقات جزئی برابر صفر و قرار دادن x_1 و x_2 به تنهایی در سمت چپ، داریم:

$$x_k^{(\ell+1)} = \frac{\sum_{j=1}^{n} \frac{w_j a_{jk}}{d'\left(X^{(\ell)}, a_j\right) d''\left(x_k^{(\ell)}, a_{jk}\right)}}{\sum_{j=1}^{n} \frac{w_j}{d'\left(X^{(\ell)}, a_j\right) d''\left(x_k^{(\ell)}, a_{jk}\right)}} \qquad (۲۱-۲)$$

که

$$d'\left(X, a_j\right) = \left(\left(\left(x_1 - a_{j1}\right)^2 + \varepsilon\right)^{p/2} + \left(\left(x_2 - a_{j2}\right)^2 + \varepsilon\right)^{p/2}\right)^{1-1/p}$$

و

$$d''\left(x_k, a_{jk}\right) = \left(\left(x_k - a_{jk}\right)^2 + \varepsilon\right)^{1-p/2}$$

و سایر مطالب همان گونه است که قبلاً برای حالت فاصله خط راست بیان شـد. در اینجـا نیـز اگـر قرار دهیم $p = ۲$ همانند روش (۲-۶) خواهد بود.

۲-۲-۲ مساله مکانیابی تک وسیله ای بر روی صفحه با تابع هدف مینیماکس

در این بخش می خواهیم نقطه جدید $X = (x_1, x_2)$ را بر روی صفحه به گونه ای مکانیابی کنیم که ماکزیمم فاصله X تا نقاط موجود داده شده a_j، $j = (1,2,...,n)$ مینیمم گردد.
معیار مینیماکس یکی از انواع معیارهای مختلف مسائل مکانیابی اسـت کـه کاربردهـای زیـادی در عرصه های عملی دارد. به طور مثال در مکانیابی خدمات اورژانسی (آمبولانس و آتش نشانی) و در مکانیابی خدمات صنعتی (تحویل فوری) که ماکزیمم تاخیر اهمیت بیشتری نسبت بـه میـانگین یـا مجموع تاخیر دارد، از این نوع مکانیابی استفاده می گردد.
شکل کلی مساله به صورت زیر می باشد:

$$\min_{(x_1,x_2)} \max_j \quad d\left(X, a_j\right) \qquad j = 1,2,...,n \qquad (۲۲-۲)$$

۲-۲-۲-۱ بررسی مساله با فاصله خط راست

تاریخچه این مساله به سال ۱۸۵۷(م.) برمی گردد که برای اولین بارتوسط سیلوسـتر[۶۱][۱۵] پایـه گذاری شد. ودر سال ۱۸۶۰(م.) روش حلی هندسی به نام روش پیرس[۶۲][۱۶] ارائـه شـد، کـه ایـن روش برای حل مساله با استفاده از دست طراحی شده بود.

[61] J.J.Sylvester

مساله (۲-۲۲) با استفاده از فاصله اقلیدسی به صورت زیر تبدیل می شود:

$$\min_{X} \max_{j} \left[(x_1 - a_{j1})^2 + (x_2 - a_{j2})^2 \right]^{1/2} \qquad j = 1,\ldots,n \qquad (۲۳-۲)$$

که در آن $X = (x_1, x_2)$ و $a_j = (a_{j1}, a_{j2})$.

این مساله با مساله زیرمعادل است :

$$\min_{(r,x_1,x_2)} r$$
$$s.t \quad \left[(x_1 - a_{j1})^2 + (x_2 - a_{j2})^2 \right]^{1/2} \le r \qquad j = (1,\ldots,n) \qquad (۲۴-۲)$$

این رابطه نشان می دهد که مساله (۲-۲۳) می تواند به صورت یافتن دایره ای با کوچکترین شعاع که شامل n نقطه موجود باشد، بیان گردد.

روش حلی که در این پایان نامه برای حل این مساله ارائه می شود، توسط جک ایلزینگا[۶۳] و دونالد هیرن[۶۴] [۱۷] مطرح شده است. که با روش پیرس نسبتاً متفاوت است و قابلیت اجرایی بیشتری بر روی کامپیوتر دارد و تعیین مینیمم دایره پوششی معادله (۲-۲۴) می باشد. واضح است که مسائل کوچک می تواند با دست سریعتر حل شود. هنگامی که نقاط a_j روی نمودار آورده شود، پیدا کردن پوسته محدب آنها ساده است. بدیهی است که دایره بهینه با دو یا سه نقطه که رئوس پوسته محدب هستند، تعریف می شود. دایره ای که قطر آن برابر فاصله نقاط مجزایی که ماکزیمم فاصله را از یکدیگر دارند باشد و تمام نقاط را نیز شامل شود، دایره جواب می باشد. به بیان دیگر سه نقطه وجود دارد که تشکیل مثلث حاده داده، به قسمی که دایره بهینه محیط بر این مثلث می باشد. دایره بهینه ممکن است از روی بیش از دو یا سه نقطه نیز بگذرد. البته ممکن است زیر مجموعه های مختلفی دایره بهینه را تعریف کند. واضح است که دایره بهینه دایره ای یکتاست. هنگامی که تعدادی نقطه وجود دارد یک تقریب مطلوب، ساختن موفقیت آمیز دایره های بزرگ و بزرگ تر است که با دو یا سه نقطه تا رسیدن به بهینگی تعریف می شود. دایره ها در هر تکرار باید اکیدا صعودی باشد و برای بدست آوردن آنها هیچ ترکیبی از نقاط نباید تکرار گردد. حال به بیان الگوریتمی که برای فاصله خط راست ارائه شده می پردازیم.

---

[62] Peirce
[63] Jack Elzinga
[64] Donald W.Hearn

الگوریتم

گام۱- دو نقطه دلخواه را انتخاب کن و به مرحله ۲ برو.

گام۲- دو نقطه را که قطر دایره را تعریف می کنند در نظر بگیر. اگر این دایره تمام نقاط را پوشش داد متوقف شو، در غیر این صورت نقطه ای بیرون از دایره انتخاب کن و همراه با دو نقطه قبلی به گام ۳ برو.

گام۳- اگر این سه نقطه یک مثلث قائمه یا منفرجه را تعریف کند، آنگاه نقطه واقع در زاویه ۹۰ ≥ درجه را رها کن و با دو نقطه باقی مانده به گام ۲ برو. در غیر این صورت اگر مثلث زاویه های اکیدا تند دارد، به گام ۴ برو.

گام۴- اگر دایره تعریف شده با سه نقطه تمام نقاط را پوشش می دهد، توقف کن. در غیر این صورت یک نقطه بیرون از دایره را انتخاب کن وD نام گذاری کن. و از بین سه نقطه ای که دایره را تعریف کرده اند، نقطه ای را که بیشترین فاصله را از D دارد، A بنام. امتداد قطر دایره فعلی که از نقطه A می گذرد، صفحه را به دو نیم صفحه تقسیم می کند. نقطه ای از دایره که D نیز در آن نیم صفحه است را B، و نقطه دیگر را C نام گذاری کن. با نقاط A، C و D به گام ۳ برو.

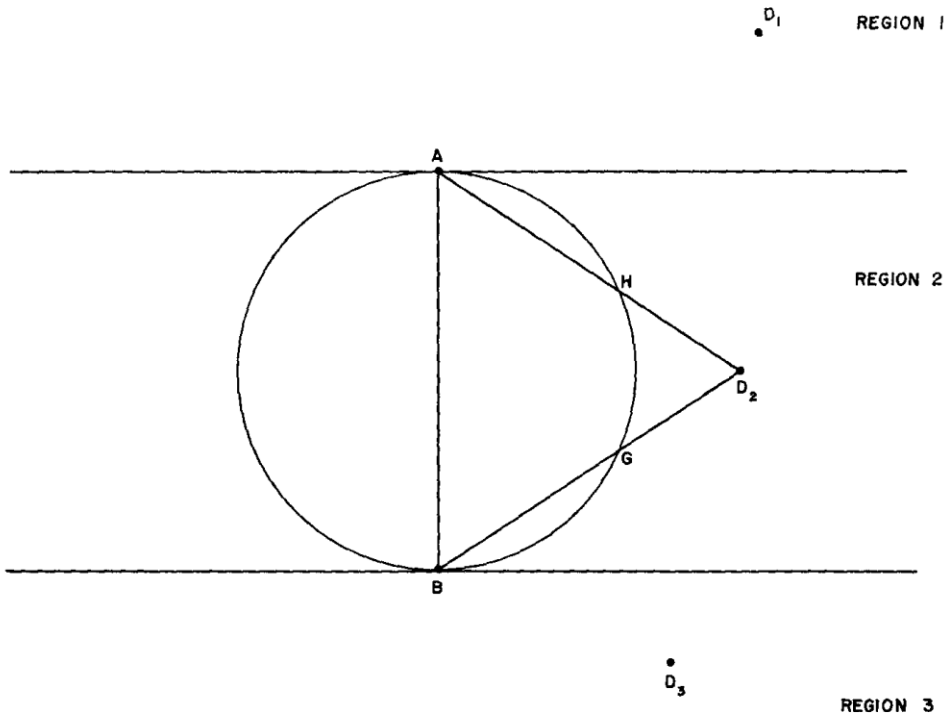
در گام ۳ اگر نقاط روی یک خط مستقیم باشند آن را مثلثی با زاویه ۱۸۰ درجه در نقطه میانی در نظر می گیریم. هنچنین توجه کنید که با در نظر گرفتن مثلث قائمه به عنوان حالت دو نقطه ای دایره، مطمئنیم که در گام ۴ نقطه B در نیم صفحه اکیدا به D متصل می شود. در عمل به نظر می رسد انتخاب دورترین نقطه بیرون از دایره معقول تر از نقاطی است که در گام ۲ و۳ انتخاب می شوند.

همگرایی این الگوریتم بر نشان دادن اینکه دایره هایی که در هر بار ساخته می شوند شعاعشان به طور یکنواخت صعود می کند، پایه گذاری شده است. چون تعدادی متناهی دایره دو نقطه ای و سه نقطه ای وجود دارد، بنابراین فرایند متناهی است.
اگر در بعضی تکرارها دایره با دو نقطه (مثل A و B) تعریف می شود، در تکرار بعدی می تواند با دو یا سه نقطه دیگر تعریف شود. با مراجعه به شکل(۲-۲) ([۱۷]) ، اگر دایره AB تمام نقاط را پوشش ندهد آنگاه نقطه بیرونی می تواند در یکی از مناطق ۱،۲ یا ۳ که در شکل نشان داده شده است، باشد. اگر در منطقه ۱ باشد می گوییم در D_1، واضح است که ABD_1 دارای یک زاویه منفرجه درA است. بنابراین گام ۳ نقطه A را رها می کند و به گام ۲ با دایره ای که دارای قطر

D_1B است، می رود. چون A نزدیک ترین نقطه به B در منطقه ۱ است پس $D_1B > AB$، بنابراین دایره بعدی بزرگتر می باشد.

اگر نقطه بیرونی در منطقه ۲ باشد مثلا در D_2، آنگاه به وضوح ABD_2 در A و B زاویه حاده دارد. از لحاظ هندسی، زاویه در D_2 برابر است با نصف کمان AB منهای نصف کمان GH. چون کمان مقابل به AB ، ۱۸۰ است درجه و کمان مقابل GH بزرگتر از صفردرجه است، مثلث ABD_2 حاده می باشد. بعلاوه دایره تعریف شده با ABD_2 بزرگتر از دایره به قطرAB است. زیرا AB یک وتر از دایره ABD_2 است نه یک قطر برای آن. ناحیه ۳ مشابه ناحیه ۱ است که تنها A با B جابه جا می شود.

به بیان دیگر دایره فعلی در بعضی تکرارها ممکن است با سه نقطه که مثلث حاده را تعریف می کند تعریف گردد. با مراجعه به گام ۴ فرض کنید A و B و C دایره فعلی را در شکل(۲-۳) ([۱۷]) با قطرهای XA و YB تعریف می کند.



شکل ۲-۲ دایره با دو نقطه AB ونقطه بیرونیD

توجه کنید که برای دایره ABC داشتن مثلث حاده B ممکن است هرجایی روی دایره بین نقاط A و X و C یا هر جایی بین X و Y باشد. اگر این دایره تمام نقاط را پوشش ندهد آنگاه نقطه خارجی

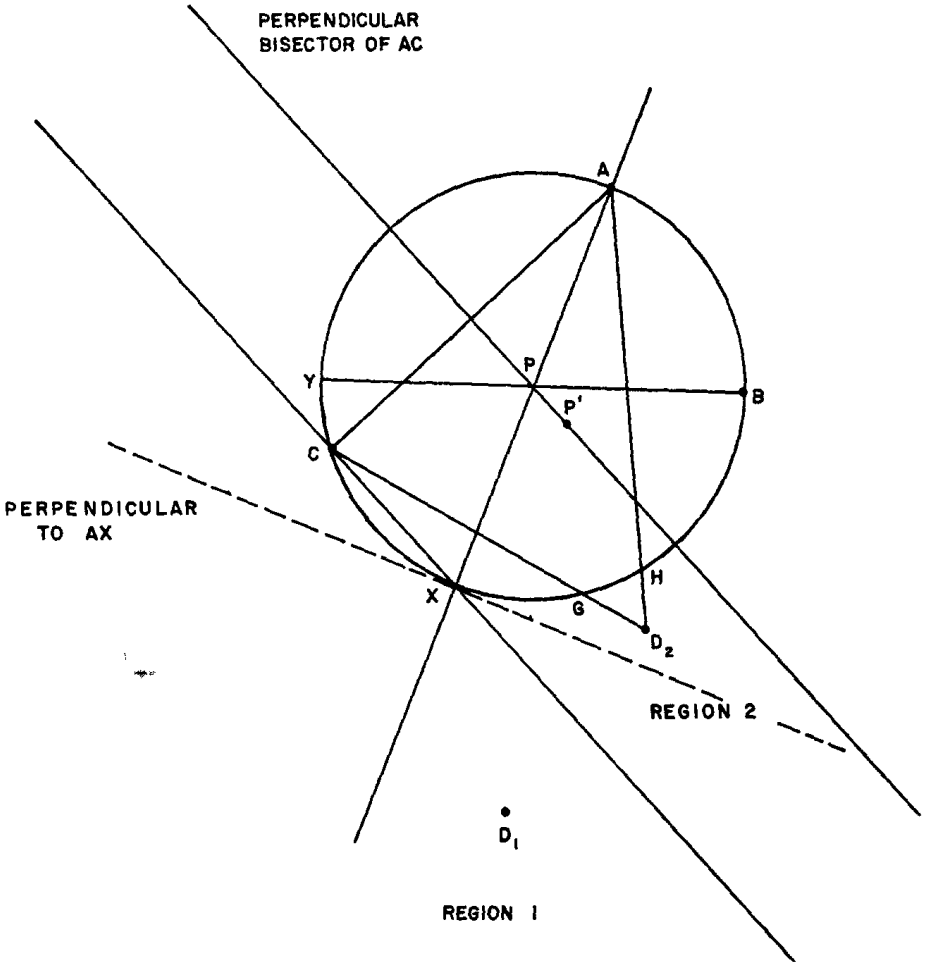
D در نیم صفحه ای که با عمودمنصف AC ایجاد می شود، با C تشکیل یک ضلع می دهد و در نیم صفحه ای که با خط APX ایجاد می شود با B مرتبط می شود.

اگر D در منطقه ۱ باشد، می گوییم D_1، آنگاه چون CX با AC تشکیل یک زاویه قائمه می دهد ، زاویه بین CD_1 و AC منفرجه است. گام۳ الگوریتم نقطه C را رها خواهد کرد و دو نقطه تعریف کننده دایره با قطر AD_1 بزرگ تر از دایره ABC است. چون $AD_1 > AX$ طبق شکل (۲-۳)، بهتر است نقطه D در منطقه ۲ باشد، می گوییم در D_2 ، آنگاه دایره بعدی با ACD_2 تعریف می شود. مثلث ACD_2 در C حاده است. زیرا CX با AC زاویه قائمه می سازد. و D_2 روی خط CX می باشد. چون $AD_2 > CD_2$ ، بنابراین $AD_2{}^2 + AC^2 - CD_2{}^2 \geq 0$ ، قانون کسینوس نشان می دهد که این مثلث در A نیز حاده است. در نهایت زاویه در D_2 حاده است، چون

$$\hat{D}_2 = \left(arc\,AC - arc\,GH\right)/2$$

و کمان AC کمتر از ۱۸۰ درجه است. دایره ACD_2 بزرگ تر از دایره ABC است. این را می توان از این حقیقت که مراکز آنها به ترتیب در P' و P قرارگرفته و بر روی عمودمنصف AC است، مشاهده کرد. چون $PD_2 > PC$ وچون $P'D_2$ باید برابر از $P'C$ باشد و P' باید به D_2 نزدیکتر و نسبت به P از C دورتر باشد، بنابراین شعاع جدید $P'C$ بزرگ تر از شعاع قبلی PC می باشد. این اثبات همگرایی را کامل می کند.

بعلاوه برای انتخاب هر دو نقطه در گام ۱، نیاز است که دو نقطه بیشترین فاصله را از هم داشته باشند. واضح است که مینیمم دایره پوششی مستلزم یک جواب اولیه مناسب است که اغلب با این دو نقطه تعریف می شود. مشکل این است که تعیین دو نقطه که بیشترین فاصله را از هم داشته باشند، نیازمند محاسبه تعداد $\left(n^2 - n\right)/2$، فاصله می باشد.



شکل ۲-۳ دایره با سه نقطه ABC و نقطه بیرونی D

مقایسه این الگوریتم با الگوریتم پیرس: الگوریتم پیرس با دایره ای که تمام نقاط را احاطه می کند، شروع می کند، سپس این دایره پیوسته کوچک می شود تا بر روی دو نقطه یا بیشتر قرار گیرد. اگر دو نقطه وجود داشته باشد که کاملا مقابل هم و بر روی قطر قرار گیرند، آنگاه فرایند متوقف می شود. و اگر سه نقطه یا بیشتر وجود داشته باشد که تشکیل یک مثلث حاده دهند نیز فرایند متوقف می گردد. در غیر این صورت روش پیرس، دایره را حول یکی از نقاط روی قطر و دور از نقطه ای که مثلث در آن زاویه منفرجه دارد، محور یابی می کند. بنابراین با این روش دایره کوچکتری ساخته می شود و فرایند تا یافتن جواب بهینه ادامه می یابد. با اینکه روش پیرس برای حل با دست به خوبی کار می کند اما اجرای آن بر روی کامپیوتر کاری پر زحمت است.

۲-۲-۲-۲ بررسی مساله با فاصله مستطیلی

از جمله کاربردهای فاصله مستطیلی برای این مساله، می توان به تحویل اورژانس، هنگامی که برای جابه جایی به یک شبکه محدود می شود اشاره کرد.
مساله (۲-۲۲) با فاصله مستطیلی به صورت زیر می باشد:

$$\min_{(x_1,x_2)} \max_{j=1,..,n} \left[\left|x_1 - a_{j1}\right| + \left|x_2 - a_{j2}\right|\right]$$

که این مساله را ریچارد فرانسیس[۶۵] [۱۸] مطرح کرد و فرمول برنامه ریزی خطی که به حل هندسی هدایت می کند ارائه داد، که در این بخش پایان نامه ارائه شده است.
کلید ساخت الگوریتم، یافتن مکان هندسی تمام نقاط هم فاصله از یک نقطه داده شده، تحت تابع فاصله مستطیلی می باشد. این مکان هندسی یک «لوزی» می باشد. یعنی مربعی که با چرخاندن ۴۵ درجه نسبت به محور مختصات بدست می آید. بنابراین مساله، یافتن کوچکترین لوزی ممکن است که تمام نقاط را پوشش دهد.

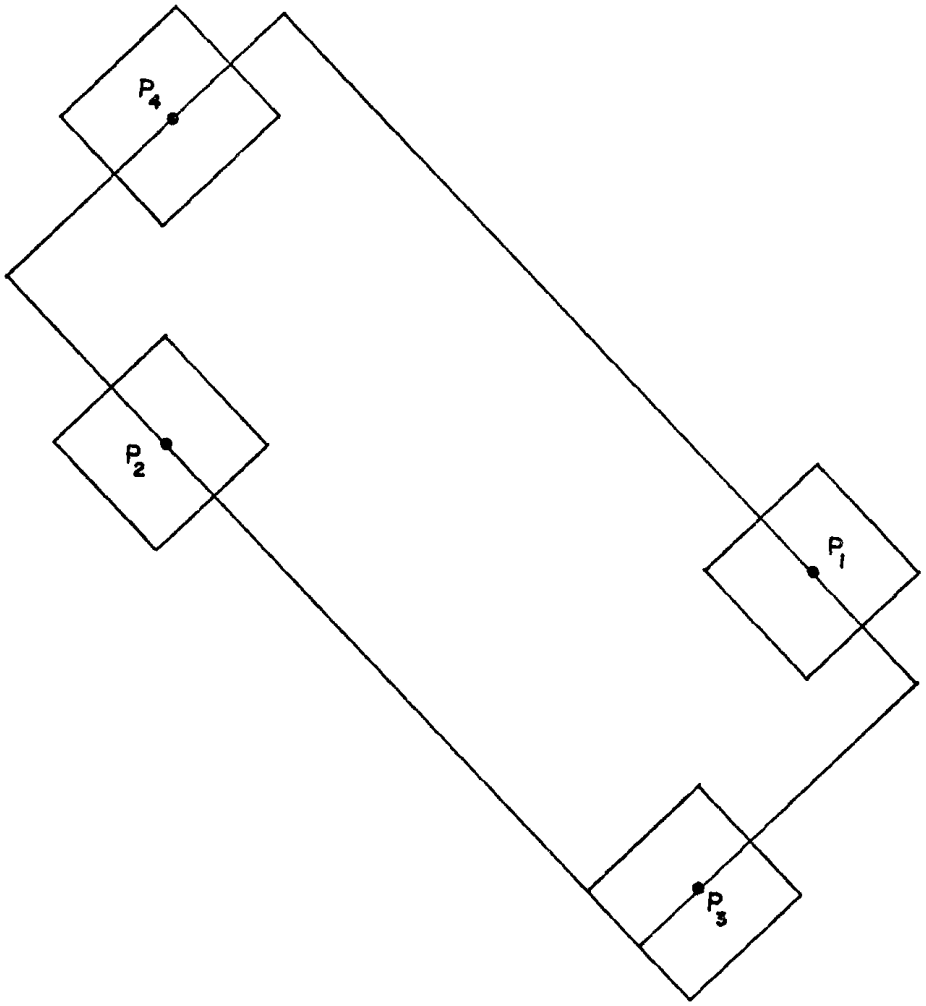
روش حل: لازم است که ابتدا، حداکثر چهار نقطه را در نظر بگیریم. که این چهار نقطه **به** صورت زیر تعریف می شوند:

$$P_1 = \max_j \left(a_{j1} + a_{j2}\right)$$
$$P_2 = \min_j \left(a_{j1} + a_{j2}\right)$$
$$P_3 = \max_j \left(a_{j1} - a_{j2}\right)$$
$$P_4 = \min_j \left(a_{j1} - a_{j2}\right)$$

حال خطی با شیب ۱- که از نقطه P_1 بگذرد و نیز خطی دیگر با شیب ۱- که از نقطه P_2 بگذرد رسم می کنیم. همچنین خطوطی با شیب های ۱+ که به ترتیب از نقاط P_3 و P_4 عبور کند، رسم می کنیم. تقاطع این چهار خط تشکیل مستطیلی می دهد که در زاویه ۴۵ درجه نسبت به محور مختصات قرار گرفته است. و هر P_i یی روی یک ضلع آن قرار گرفته است.
می خواهیم کوچکترین لوزی پوششی برای این چهار نقطه بسازیم و نشان دهیم که نقاط باقیمانده نیز پوشش داده می شوند. درشکل(۲-۴) ([۱۷]) چهارنقطه عامل P_i ، یک مستطیل و

[65] Richard L. Francis

حول هر نقطه یک لوزی را نشان می دهد. لوزی های هر نقطه نشان دهنده مکان هندسی تمام نقاط با فاصله d ($d < d^*$) از آن نقطه است که d^* فاصله بهینه است. به وضوح هیچ نقطه داخلی یک لوزی نمی تواند جواب باشد زیرا ماکزیمم فاصله مینیمم نشده است.

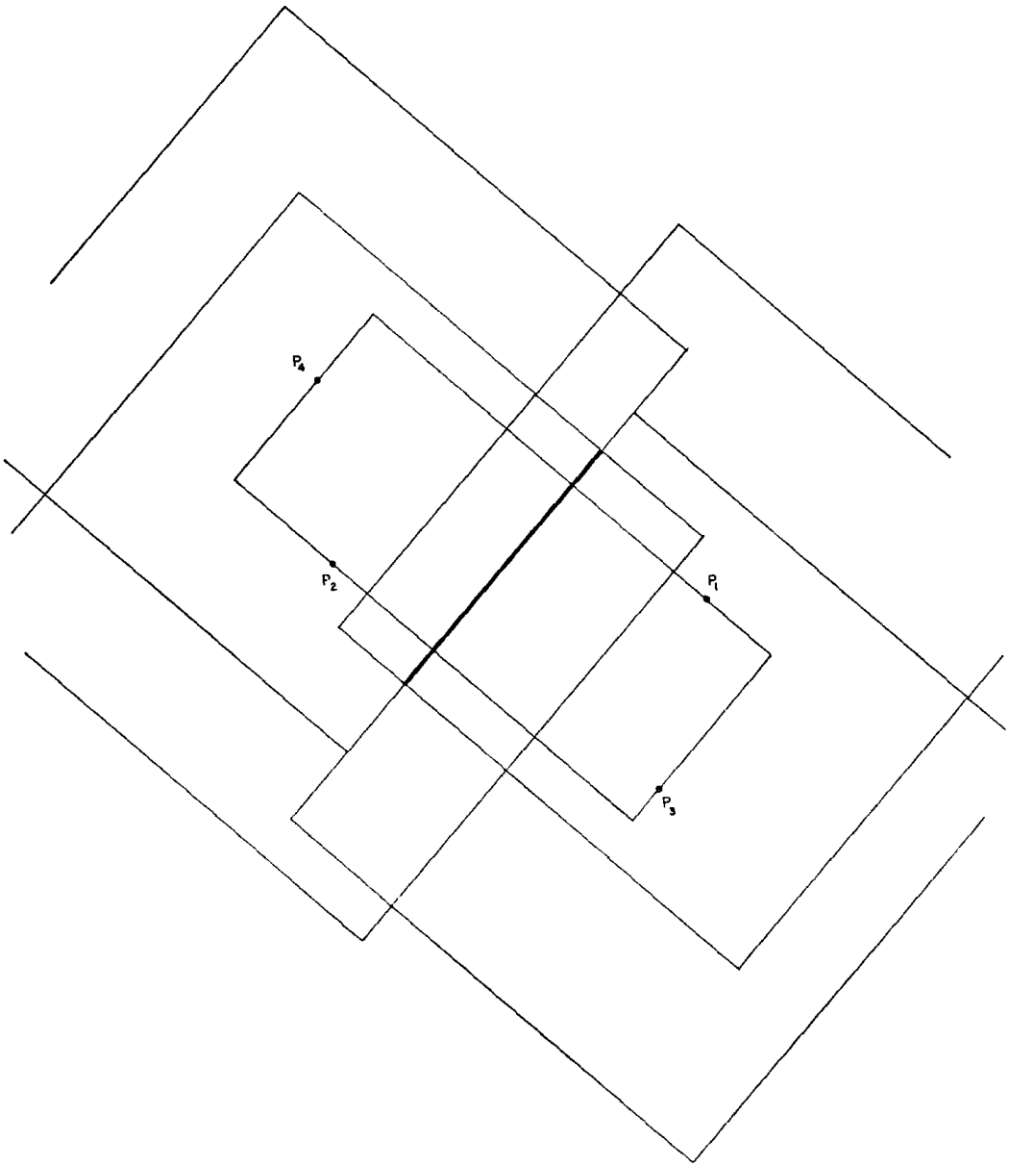


شکل۲-۴ مستطیل پوششی و لوزی های هم فاصله

هنگامیکه d افزایش یابد، لوزی ها حول مراکزشان بسط می یابند. جواب مساله (لوزی پوششی) هنگامی بدست می آید که اولین نقطه مشترک چهار لوزی حاصل می شود. هر نقطه ای که روی این فصل مشترک قرار دارد جواب مساله می باشد. این جواب در شکل (۲-۵) ([۱۷]) نشان داده شده است.

مشاهده می کنیم که جواب، قسمتی از عمودمنصف ضلع بزرگتر مستطیل است. بنابراین جواب، به مستطیل ساخته شده، فقط وابسته است و روی مکان خاصی از اضلاع مستطیل نمی باشد. یعنی اینکه اگر نقطه P_1 هرجای ضلع قرار گرفته باشد، مکان هندسی نقاط تغییر نمی کند و این مطلب

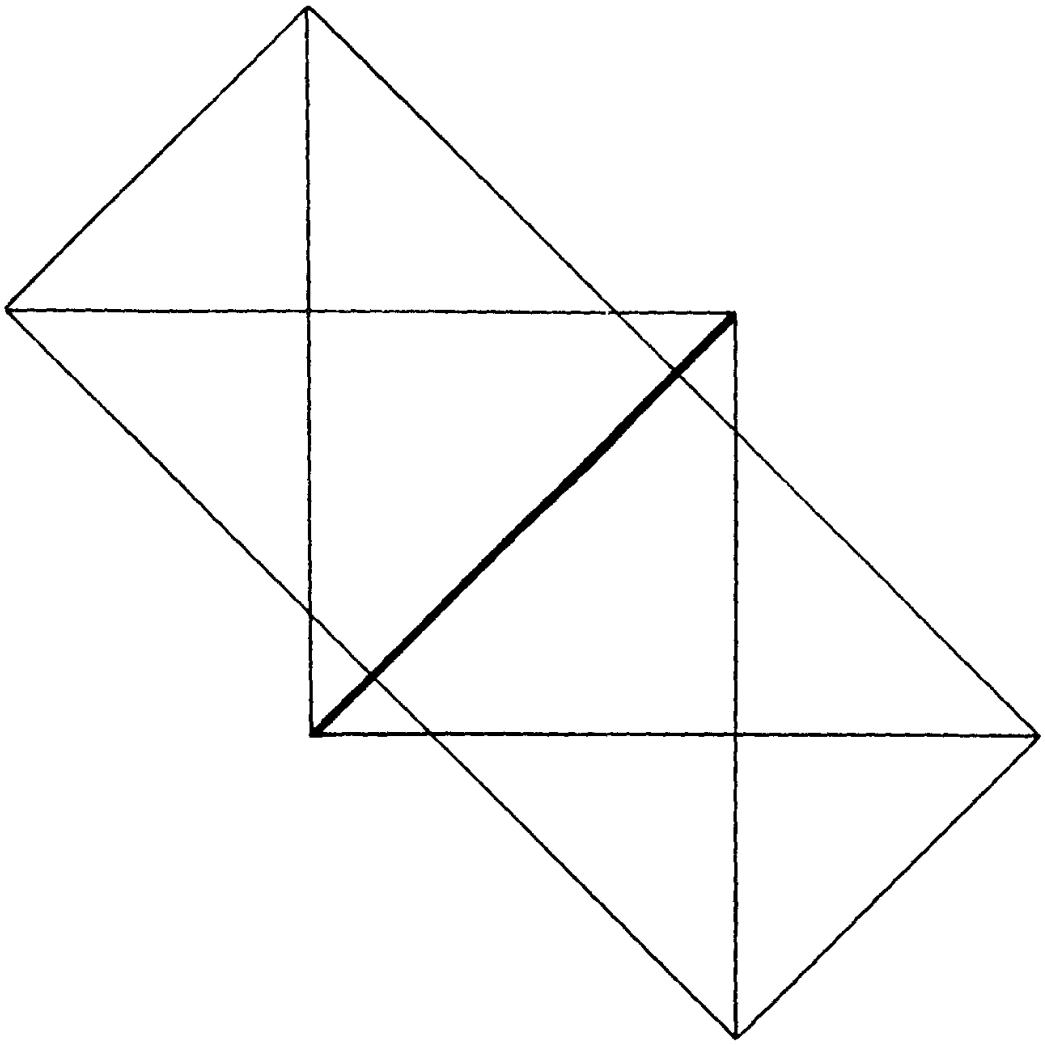
برای سایر نقاط P_i صادق می باشد. به این ترتیب جوابی که ساخته ایم، همان مینیمم لوزی پوششی برای تمام نقاط روی محیط مستطیل است. سایر نقاط باقی مانده درون مستطیل قرار دارند، فاصله هر یک از آنها تا هر نقطه جواب باید کمتر از d^* باشد، زیرا هر نقطه درون مستطیل، روی مسیر به طول کمتر از d^* که فاصله هر نقطه روی محیط مستطیل تا نقطه جواب است، قرار دارد.



شکل ۲-۵ توسعه لوزی های تعریف کننده مکان هندسی مینیماکس

ساختار جواب مینیماکس نشان می دهد که مجموعه نقاط جواب می تواند با رسم عمودمنصف بزرگترین ضلع مستطیل و قطع آن با خطوط عمودی و افقی ای که از انتهای کوتاهترین اضلاع

مستطیل کشیده شده اند، پیدا شود، به صورتی که در شکل (۲-۶) ([۱۷]) نشان داده شـده اسـت. جبر ساده که روی شکل (۲-۶) پایه گذاری شده، قضیه زیر را به ما می دهد.



شکل ۲-۶ مکان هندسی مینیماکس که با مستطیل پوششی تعریف می شود

قضیه([۱۸]): مینیمم لوزی پوششی مساله $\min_{(x_1,x_2)} \max_{j=1,..,n} \left[\left|x_1 - a_{j1}\right| + \left|x_2 - a_{j2}\right|\right]$ را در نظر بگیریـد. فـــــرض کنیـــــد: $\max\left(a_{j1} + a_{j2}\right) = u_1$ ، $\min\left(a_{j1} + a_{j2}\right) = u_2$ و $\max\left(a_{j1} - a_{j2}\right) = v_1$ ، $\min\left(a_{j1} - a_{j2}\right) = v_2$. بعـــــلاوه فـــــرض کنیـــــد $\max\left\{(u_1 - u_2),(v_1 - v_2)\right\} = s_1 - s_2$ و $\min\left\{(u_1 - u_2),(v_1 - v_2)\right\} = t_1 - t_2$ باشـد، آنگـاه فاصـله مینیمـاکس d^* برابـر اسـت بـا $(s_1 - s_2)/\sqrt{2}$ و مکان مینیماکس $\left(x_1^*, x_2^*\right)$ است، اگر و فقط اگر

$$\left(x_1^*, x_2^*\right) \in \left\{ \left(x_1, x_2\right) \middle| \begin{array}{l} x_1 = \lambda\left(s_2 + t_1\right)/2 + \left(1-\lambda\right)\left(s_1 + t_2\right)/2, \\ x_2 = \lambda\left(u_1 - v_1\right)/2 + \left(1-\lambda\right)\left(u_2 - v_2\right)/2, \quad 0 \le \lambda \le 1 \end{array} \right\}$$

فصل۳
مساله مکانیابی تک وسیله ای بر روی خط

۳-۱ مقدمه

در این فصل می خواهیم مساله مکانیابی تک وسیله ای بر روی صفحه، با این محدودیت که مکان جدید روی خط مشخص L قرار بگیرد را بررسی کنیم. این مساله حالت خاصی از مساله مکانیابی تک وسیله ای بر روی صفحه می باشد. که در عمل می تواند کاربردهای زیادی داشته باشد. از جمله کابردهای این مساله با تابع هدف کمترین مجموع می توان به ساختن یک سد بر روی رود، به قسمی که مجموع فواصل وزن دار آن تا زمین های کشاورزی، یا روستاها و شهر های اطراف آن کمترین مقدار گردد. و همچنین به خدمات اورژانس یا آتش نشانی یا خط تولید یک کارخانه و ... ، اگر تابع هدف مساله مینیماکس باشد، اشاره کرد.

۳-۲ مساله مکانیابی تک وسیله ای بر روی خط با تابع هدف کمترین مجموع

فرض کنید تعدادی نقطه بر روی صفحه قرار دارد. می خواهیم مکان نقطه جدید را به گونه ای تعیین کنیم که اولا مکان جدید روی خط مشخص L قرار بگیرد و ثانیا مجموع فاصله های وزن دار آن تا نقاط موجود، کمینه گردد.

فرم کلی مساله به صورت زیر می باشد:

$$\min_{X\in L} \quad W(X)=\sum_{j=1}^{n} w_j \ell_p(X,a_j) \qquad (۱-۳)$$

که در آن

n: تعداد نقاط موجود در صفحه،

w_j: هزینه انتقال از نقطه جدید تا نقاط موجود a_j و ($w_j > 0$)،

$a_j = (a_{j1}, a_{j2})$: موقعیت نقطه موجود a_j روی صفحه،

$X = (x_1, x_2)$: موقعیت نقطه جدید روی خط،

$\ell_p(X,a_j)$: فاصله بین نقطه جدید با نقاط موجود a_j

برای راحت تر شدن مساله، فرض می کنیم که خط L، منطبق بر محور x ها باشد. در غیر این صورت می توان با یک انتقال، خط L را بر روی محور x ها انتقال داد و مطابق با این انتقال، نقاط موجود a_j را نیز منتقل کرد. بنابر این مختصات نقطه جدید به صورت $X = (x_1, 0)$ می باشد. همچنین در این فصل فرض بر این است که هیچ یک از نقاط موجود، بر روی خط L قرار ندارند.

طبق آنچه که در فصل ۲ گفته شد، می دانیم که جواب مساله درون پوسته محدب نقاط موجود، می باشد. حال اگر خط L، درون پوسته محدب نقاط داده شده باشد، آنگاه جواب بهینه مساله روی آن قسمت از خط که درون این پوسته است، می باشد. و اگر خط L بیرون پوسته محدب نقاط موجود باشد، آنگاه با تصویر کردن نقاط موجود بر روی خط L، پاره خط مورد نظر به دست می آید.

اگر این پاره خط را $[a,b]$ بنامیم، آنگاه $a=\min_j\{a_{j1}\}$ و $b=\min_j\{a_{j2}\}$. بنا براین مساله (۳-۱) را می توان به صورت زیر نوشت:

$$\min_{X\in[a,b]} \quad W(X)=\sum_{j=1}^{n} w_j \ell_p(X,a_j) \qquad (۲-۳)$$

۳-۲-۱ بررسی مساله با فاصله خط راست

برای حل مساله با فاصله اقلیدسی، مساله (۳-۲) به صورت زیر تبدیل خواهد شد:

$$\min_{X\in[a,b]} \quad W(X)=\sum_{j=1}^{n} w_j\left((x_1-a_{j1})^2+(0-a_{j2})^2\right)^{1/2} \qquad (۳-۳)$$

چون $W(X)$ در رابطه (۳-۳) فقط به متغیر x_1 وابسته است بنابراین می توان مساله (۳-۳) را به صورت زیر نوشت:

$$\min_{x_1\in[a,b]} \quad W(x_1)=\sum_{j=1}^{n} w_j\left((x_1-a_{j1})^2+a_{j2}^2\right) \qquad (۴-۳)$$

برای حل مساله بررسی می کنیم که آیا تابع هدف (۳-۴) محدب است یا نه. چون وزن ها مثبت هستند بنابراین به ازای هر j، $w_j\left((x_1-a_{j1})^2+a_{j2}{}^2\right)^{1/2}$ یک تابع محدب می باشد. و از آنجایی که مجموع توابع محدب، محدب می باشد. بنابراین $W(x_1)$ محدب می باشد.

روش های گوناگونی که برای حل مسائل بهینه سازی محدب وجود دارد را می توان برای حل این مساله به کار برد. از جمله این روش ها می توان به روش وایس فیلد اشاره کرد. همچنین از روش های حل جستجوی خطی[۶۶] نیز می توان استفاده نمود.

۳-۲-۱-۱ حل مساله به روش وایس فیلد

مطابق با مطالبی که درباره روش وایس فیلد در فصل دوم گفته شد، معادله اکسترمالی که می تواند بهینه سراسری را برای این مساله تولید کنند به صورت زیر می باشد:

[66] Line search

$$\frac{d\,W(x_1)}{d\,x_1}=\sum_{j=1}^{n}\frac{w_j\left(x_1-a_{j1}\right)}{\left(\left(x_1-a_{j1}\right)^2+{a_{j2}}^2\right)^{1/2}}=0 \qquad (۵-۳)$$

با باز نویسی معادله (۳-۵)، خواهیم داشت:

$$x_1=\frac{\sum_{j=1}^{n}\frac{w_j a_{j1}}{\left(\left(x_1-a_{j1}\right)^2+{a_{j2}}^2\right)^{1/2}}}{\sum_{j=1}^{n}\frac{w_j}{\left(\left(x_1-a_{j1}\right)^2+{a_{j2}}^2\right)^{1/2}}} \qquad (۶-۳)$$

فرض کنید ℓ تکرار کامل داریم و مکان $\left(x_1^{(\ell)},0\right)$ را به دست آورده ایم، آنگاه با استفاده از معادله (۳-۵) برای $(\ell+1)$امین تکرار داریم:

$$x_1^{(\ell+1)}=\frac{\sum_{j=1}^{n}\frac{w_j a_{j1}}{\left(\left(x_1^{(\ell)}-a_{j1}\right)^2+{a_{j2}}^2\right)^{1/2}}}{\sum_{j=1}^{n}\frac{w_j}{\left(\left(x_1^{(\ell)}-a_{j1}\right)^2+{a_{j2}}^2\right)^{1/2}}} \qquad (۷-۳)$$

قبل از شروع تکرارها به یک نقطه آغازین $\left(x_1^{(0)},0\right)$ نیازمندیم. یک راه انتخاب نقطه اولیه، انتخاب جواب مساله فاصله مربعی اقلیدسی مساله (۳-۱) است. که مشابه مساله (۳-۳) است با این تفاوت که فاصله $\ell_2\left(X,a_j\right)$، مربعی می باشد. که مرکز گرانشی مکانیابی، جواب بهینه مساله با فاصله مربعی اقلیدسی $\min W(X)=\sum_{j=1}^{n}w_j\left[\left(x_1-a_{j1}\right)^2+\left(0-a_{j2}\right)^2\right]$ است.

بنابراین نقطه شروع برای رویه (۳-۷) به صورت زیر است:

$$x_1^{(0)}=\frac{\sum_{j=1}^{n}w_j a_{j1}}{\sum_{j=1}^{n}w_j} \qquad (۸-۳)$$

که با این رویه، تکرارها به مکان بهینه $\left(x_1^*,o\right)$ میل خواهد کرد.

خاتمه این روش، معیار توقف آن، یعنی کران پایین مقدار بهینه $W(X)$ بر روی خط L می باشد. این کران پایین دائماً در طی تکرارها به روز می شود. برای بدست آوردن این کران نیازمند گزاره زیر هستیم.

گزاره ۳-۱[دامنه پوسته محدب]: جواب بهینه مساله (۳-۱) باید درون بازه $[a,b]$ (پوسته محدب تصویر نقاط موجود مکانیابی بر روی خط L) قرار گیرد.

می توان نشان دادکه نقطه شروع، با معادله (۳-۸) در این پوسته تعریف می شود. بعلاوه به غیر از نقطه شروع استفاده شده در رویه (۳-۷)، نقاط بعدی نیز در پوسته محدب خواهد بود. یک منطق هندسی برای روش وایس فیلد مشابه مطالب فصل ۲ ارائه می گردد.

۳-۲-۱-۲ حل مساله به روش جستجوی خطی

یکی از روش های بهینه سازی نامقید استفاده از روش های جستجوی خطی است. که این جستجوی خطی معادل است با مینیمم سازی یک تابع محدب یک یا چند متغیره بدون محدودیت یا با محدودیت های ساده شبیه کران های بالا و پایین متغیر.
دو نوع جستجوی خطی وجود دارد. یکی جستجوی خطی بدون استفاده از مشتقات و دیگری جستجوی خطی با استفاده از مشتقات می باشد. که برای هر یک از این دو نوع، روش های متعددی برای مینیمم سازی توابع محدب وجود دارد. روش های جستجوی خطی بدون استفاده از مشتق مثل جستجوی یکنواخت، جستجوی دوبخشی، روش تقسیم طلایی و روش فیبوناتچی می باشد. روش های جستجوی خطی با استفاده از مشتق عبارتند از روش دو بخشی و روش نیوتن. چون در عمل معمولا روش تقسیم طلایی بیشتر به کار می روند، ما نیز برای حل مساله و از این روش استفاده می کنیم.

۳-۲-۱-۲-۱ حل مساله به روش تقسیم طلایی

مساله زیر را در نظر بگیرید:

$$\min_{a\le\lambda\le b} \theta(\lambda)$$

چون مکان دقیق جواب مشخص نیست، بازه $[a,b]$ را غیر قطعی می گویند. در طول فرایند جستجو این بازه در هر مرحله کوچک می شود تا به دقت داده شده برای جواب برسیم.

قضیه ۳-۱: فرض کنید $\theta : R \to R$ روی بازه $[a,b]$ محدب باشد. همچنین فرض کنید $\lambda, \mu \in [a,b]$ به گونه ای باشند که $\lambda < \mu$. در این صورت اگر $\theta(\lambda) < \theta(\mu)$ ، آنگاه برای هر $z \in (\mu, b]$ داریم: $\theta(z) > \theta(\lambda)$ و اگر $\theta(\lambda) > \theta(\mu)$ ، آنگاه برای هر $z \in [a, \lambda)$ خواهیم داشت: $\theta(z) > \theta(\mu)$.

در روش تقسیم طلایی، در هر تکرار نسبت طول بازه جدید به طول بازه قبلی عدد طلایی می باشد که عدد طلایی ریشه معادله $x^2 + x - 1 = 0$ بوده و برابراست با: $\alpha = 0.6181$

در تکرار کلی k ام از این روش، فرض می کنیم بازه $[a_k,b_k]$ بازه غیر قطعی باشد، با استفاده از قضیه (۳-۱) می توان بازه بعدی را تعیین کرد.
بازه جدید غیر قطعی $[a_{k+1},b_{k+1}]$ توسط $[\lambda_k,b_k]$ تعیین می شود هرگاه $\theta(\lambda_k)>\theta(\mu_k)$ و توسط $[a_k,\mu_k]$ تعیین می شود هرگاه $\theta(\lambda_k)<\theta(\mu_k)$.

الگوریتم

گام۱: $[a_1,b_1]$ را بازه آغازین، α را برابر عدد طلایی، $\lambda_1=a_1+(1-\alpha)(b_1-a_1)$ و $\mu_1=a_1+\alpha(b_1-a_1)$ قرار دهید و L را دقت بازه انتخاب کنید.

گام ۲: اگر $b_k-a_k<L$ باشد توقف کنید. جواب در بازه $[a_k,b_k]$ با دقت L قرار دارد.

گام ۳: اگر $\theta(\lambda_k)>\theta(\mu_k)$ باشد، قرار دهید: $a_{k+1}=\lambda_k$ و $b_{k+1}=b_k$ و همچنین مقدار جدید $\mu_{k+1}=a_{k+1}+\alpha(b_{k+1}-a_{k+1})$ و $\lambda_{k+1}=\mu_k$. در غیر این صورت قرار دهید: $a_{k+1}=a_k$ و $b_{k+1}=\mu_k$ و $\mu_{k+1}=\lambda_k$ و $\lambda_{k+1}=a_{k+1}+(1-\alpha)(b_{k+1}-a_{k+1})$

گام ۴: $k=k+1$ قرار دهید و به گام ۲ بروید.

مثال۳-۱: ده نقطه (۱،۱), (۱۲،۲) , (۱۱،۴), (۱،۵)، (۵،۶)، (۶،۷)، (۱،۸)، (۱۰،۸)، (۱۰،۱)، (۱،۱۰) به عنوان نقاط موجود, به ترتیب با وزن های ۱، ۵، ۴، ۴،۷، ۸، ۷، ۸، ۱و۱داده شده است. با استفاده از فاصله خط راست مکان بهینه ای برای نقطه جدید روی خط x=0، بیابید.

جواب: ابتدا پوسته محدب نقاط داده شده را روی خط x=0 ، جهت به دست آوردن بازه $[a,b]$ به دست می آوریم و سپس تابع هدف مثال را با مطابق فرمول (۳-۴) می نویسیم. با تصویر نقاط بر روی خط x=0 ، بازه $[1,12]$ به عنوان بازه مورد نظر به دست می اید. بنابراین تابع هدف به صورت زیر می شود:

$$\min_{x\in[1,10]} W(x)=\sum_{j=1}^{4} w_j\left((x-a_{j1})^2+a_{j2}^2\right)^{1/2}$$
$$=1\left((x-1)^2+1\right)^{1/2}+5\left((x-12)^2+4\right)^{1/2}+4\left((x-11)^2+16\right)^{1/2}+\cdots+1\left((x-1)^2+100\right)^{1/2}$$

اگر $L=10^{-6}$ در نظر بگیریم، آنگاه با استفاده از جستجوی تقسیم طلایی جواب برابر است با: $x=1$.

۳-۲-۲ بررسی مساله با فاصله مستطیلی

برای حل مساله با فاصله مستطیلی، مساله (۳-۲) به صورت زیر تبدیل خواهد شد:

$$\min_{X\in[a,b]} \quad W(X)=\sum_{j=1}^{n} w_j\left(\left|x_1-a_{j1}\right|+\left|0-a_{j2}\right|\right) \qquad (۱۱-۳)$$

چون $W(X)$ در رابطه (۳-۱۱) فقط به متغیر x_1 وابسته است بنابراین می توان مساله (۴-۱۱) را به صورت زیر نوشت:

$$\min_{x_1\in[a,b]} \quad W\left(x_1\right)=\sum_{j=1}^{n} w_j\left(\left|x_1-a_{j1}\right|+\left|a_{j2}\right|\right) \qquad (۱۲-۳)$$

برای حل مساله بررسی می کنیم که آیا تابع هدف (۳-۱۲) محدب است یا نه. چون وزن ها مثبت هستند بنابراین، به ازای هر j، $w_j\left(\left|x_1-a_{j1}\right|+\left|a_{j2}\right|\right)$ یک تابع محدب می باشد. و از آنجایی که مجموع توابع محدب، محدب می باشد. بنابراین $W\left(x_1\right)$ محدب می باشد.

روش های گوناگونی را می توان برای حل این مساله به کار برد. ساده ترین روش برای این مساله به صورت زیر می باشد.

حل مساله (۳-۱۲): برای تحلیل آسان تر حل، علامت گذاری (۳-۱۲) را تغییر می دهیم و مقادیر a_{j1} را به ازای هر j دوباره مرتب می کنیم، به صورتی که $a_{(1)1}<a_{(2)1}<a_{(3)1}<\ldots<a_{(n_k)1}$ باشد و فرض می کنیم که $w_1^1,w_2^1,\ldots,w_{n_k}^1$ مطابق وزن های مثبت در بعد x_1 باشد. توجه کنید که حال، تعداد n_k مختصات وجود خواهد داشت که $n_k<n$.

مثلاً فرض کنید اگر $a_{11}=a_{31}=4$ و $w_1=3$ و $w_3=2$ باشد، آنگاه می توان مجموع جملات $3\left|x_1-4\right|+\left|x_1-4\right|$ را به یک جمله $5\left|x_1-4\right|$ تبدیل کرد.

بنابراین می توان دنباله ای از $a_{(j)1}$ ها را ساخت که برحسب مقدار اکیداً نزولی هستند. و در نتیجه آن، ممکن است $a_{(j)1}$ ها کمتر از a_{j1} ها گردند.

مساله (۳-۱۲) به صورت زیر نوشته می شود:

$$\min_{x_1\in[a,b]} \quad W\left(x_1\right)=\sum_{j=1}^{n} w_j\left(\left|x_1-a_{(j)1}\right|+\left|a_{(j)2}\right|\right) \qquad (۱۳-۳)$$

حال تابع $W\left(x_1\right)$ فرمی مناسب برای محاسبه مشتق دارد و می توان نوشت:

$$W'(x_1) = -\sum_{j=1}^{n_k} w_j^1 \qquad x_1 < a_{(1)1} \qquad (۱۴\ \text{a-}۳)$$

$$W'(x_1) = \sum_{j=1}^{t} w_j^1 - \sum_{j=t+1}^{n_k} w_j^1 \qquad a_{(t)1} < x_1 < a_{(t+1)1} \qquad (۱۴\ \text{b-}۳)$$

$$W'(x_k) = \sum_{j=1}^{n_k} w_j^1 \qquad x_1 > a_{(n_k)1} \qquad (۱۴\ \text{c-}۳)$$

مشاهده می کنیم که شیب تابع $W(x_1)$ از قطعات خطی ساخته شده است و تنها در نقاط $a_{(j)1}$ تغییر می کند. در مجموع می توان گفت $W(x_1)$ تابعی پیوسته-محدب و تکه ای-خطی است که مشتق اول آن در نقاط a_{j1} ناپیوسته است. چون شیب $W(x_1)$ در معادله (۴-۱۴) با افزایش t، افزایش می یابد، بنابراین مینیمم $W(x_1)$ هر جا که شیب تابع از منفی به مثبت یا در بعضی نقاط از منفی به صفر تغییر می کند، اتفاق می افتد. و در حالتی دیگر مینیمم تابع روی یک فاصله از x_1 اتفاق می افتد، هرچند که حداقل یک نقطه $a_{(t)1}$ باید مینیمم کننده $W(x_1)$ باشد. همانند فصل ۲ می توان گزاره ای به صورت زیر برای مساله بیان کرد.

گزاره ۳-۳ [شرایط برای مینیمم $W(x_1)$]: فرض کنید

$$\sum_{j=1}^{t-1} w_j^1 - \sum_{j=t}^{n_k} w_j^1 < 0 \qquad (۱۵\ \text{a-}۳)$$

و

$$\sum_{j=1}^{t} w_j^1 - \sum_{j=t+1}^{n_k} w_j^1 \geq 0 \qquad (۱۵\ \text{b-}۳)$$

در بعضی نقاط t^* صادق اند،

اگر شرط (۱۵ b-۳) نامساوی اکید باشد، آنگاه $x_1^* = a_{(t^*)1}$

و اگر شرط(۱۵ b-۳) نامساوی باشد، آنگاه $x_1^* \in \left[a_{(t^*)1}, a_{(t^*+1)1}\right)$ می شود.

حال اگر قرار دهیم: $C = \sum_{j=1}^{n_k} w_j^1$

همانند مطالب فصل ۲ شرایط گزاره (۴-۱۵) به صورت زیر خواهد بود:

$$-C + 2\sum_{j=1}^{t-1} w_j^1 < 0 \qquad (۱۶\ \text{a-}۳)$$

$$-C + 2\sum_{j=1}^{t} w_j^1 \geq 0 \qquad (۱۶\ \text{b-}۳)$$

حال نامساوی های (۳-۱۶) یک روش محاسباتی برای حل مساله پیشنهاد می کند. از $t=1$ شروع می کنیم و دو برابر مجموع وزن های نقاط $a_{(t)1}$ را تا اولین نقطه ای که با C برابر شود یا از آن بیشتر گردد، محاسبه می کنیم. حال نقطه بهینه x_1^* همان اولین نقطه می باشد.

مثال(۳-۲): چهار نقطه (۱,۱) ، (۲,۴) ، (۲,۳) و (۴,۲) را به ترتیب با وزن های ۲ ، ۳ ، ۱ و ۲ در نظر بگیرید و با استفاده از فاصله مستطیلی مکان بهینه ای برای نقطه جدید بر روی خط x=0 به دست آورید.

حل:

جدول ۳-۱ مثال (۳-۲)

ترتیب در بعد x_1		مختصات و وزن های اصلی			
w_j^1	$a_{(j)1}$	w_j	a_{j2}	a_{j1}	j
۲	۱	۲	۱	۱	۱
۴	۲	۳	۴	۲	۲
۲	۴	۱	۳	۲	۳
		۲	۲	۴	۴

با استفاده از این جدول محاسبات زیر را انجام می دهیم:

$$C=\sum_{J=1}^{n_k} w_j^1 = ۲+۴+۲=۸$$

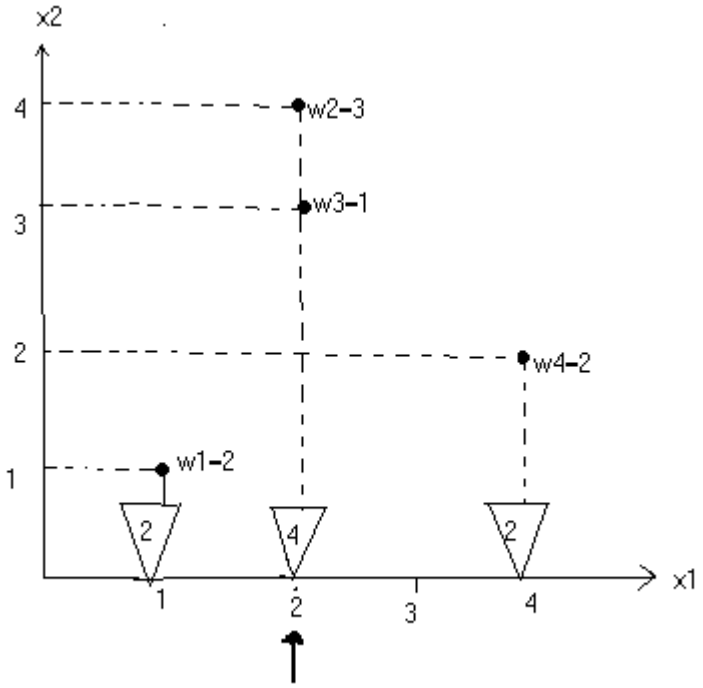
حال برای یافتن x_1^* محاسبات زیر را داریم:

$$-C+2\sum_{j=1}^{1} w_j^1 = ۸+۴=-۴$$

$$-C+2\sum_{j=1}^{2} w_j^1 = \ ۴+۱۲=۴>۰$$

چون شرط (۳-۱۶ b) به صورت اکید برقرار است، بنابراین $x_1^* = a_{(2)1} = 2$

اما روش راحت تری برای حل این مساله با استفاده از نمودار به صورت زیر وجود دارد:
وزن ها را بر روی تصویر نقاط، در محور x_1 و x_2 می نویسیم (شکل۳-۱). محور x_1 را در آنجایی که وزن ها نصف می شود، به دو قسمت تقسیم می کنیم. که در مثال (۳-۲) این نقطه $x_1 = 2$ می باشد. بنابراین مقدار جواب بهینه ای که به دست می آید، در شرایط (۳-۱۵) و(۳-۱۶) صدق می کند.



شکل ۳-۱ نمایش وزن میانی

۳-۲-۳ بررسی مساله با فاصله ℓ_p

مساله (۳-۲) با فاصله ℓ_p به صورت زیر است:

$$\min_{x_1\in[a,b]} \; W(x_1)=\sum_{j=1}^{n} w_j\left(\left|x_1-a_{j1}\right|^p+\left|a_{j2}\right|^p\right)^{1/p} \qquad (۱۷-۳)$$

به وضوح هر جمله $W(x_1)$ در (۳-۱۷) محدب است همچنین با استفاده از این حقیقت که مجموع توابع محدب، محدب است، نتیجه می گیریم که $W(x_1)$ تابعی محدب می باشد. از این رو یک مینیمم موضعی، مینیمم سراسری می باشد.

هر جمله $|y|$ در مساله (۳-۱۷) را با $(y^2+\varepsilon)^{1/2}$ که درآن ε یک عدد مثبت کوچک است، تعویض می کنیم. این تقریب همیشه بزرگتر از مقدار اصلی می باشد، اما هنگامی که $\varepsilon\to 0$، این تقریب نیز به جمله اصلی میل خواهد کرد. و مساله (۳-۱۸) با رابطه زیر تقریب زده می شود:

$$\min_{x_1\in[a,b]} WH(x_1)=\sum_{j=1}^{n} w_j\left[\left((x_1-a_{j1})^2+\varepsilon\right)^{p/2}+\left(a_{j2}^2+\varepsilon\right)^{p/2}\right]^{1/p} \qquad (۱۸-۳)$$

ملاحظه می کنیم که $WH(x_1)$ اکیداً محدب است و تمام مشتقات آن در هر نقطه ای پیوسته است. بنابراین $WH(x_1)$ یک کاندید خوش تعریف برای نزول غیر خطی کلی و الگوریتم هایی است که اگر تابع یکنواخت یا اکیداً یک کوهانه باشد، به مینیمم کننده همگرا می شود.

سوالی که پیش می آید این است که اگر نقطه ای که $WH(x_1)$ را مینیمم می کند یافتیم، آیا این نقطه $W(x_1)$ را نیز مینیمم می کند؟ همانند فصل ۲ می توان نشان داد که می توان مینیمم کننده $W(x_1)$ را با مینیمم کننده $WH(x_1)$ با انتخاب مقادیر کوچک تقریب زد.

روش وایس فیلد نیز می تواند برای حل مساله (۳-۱۷) مورد استفاده قرار گیرد. با قرار دادن مشتقات جزئی برابر صفر و قرار دادن x_1 در سمت چپ، داریم:

$$x_1^{(\ell+1)} = \frac{\sum_{j=1}^{n} \frac{w_j a_{jk}}{d'\left(x_1^{(\ell)}, a_j\right) d''\left(x_1^{(\ell)}, a_{j1}\right)}}{\sum_{j=1}^{n} \frac{w_j}{d'\left(x_1^{(\ell)}, a_j\right) d''\left(x_1^{(\ell)}, a_{j1}\right)}} \qquad (۲۱-۳)$$

که

$$d'(x_1, a_j) = \left(\left((x_1 - a_{j1})^2 + \varepsilon\right)^{p/2} + \left((a_{j2})^2 + \varepsilon\right)^{p/2}\right)^{1-1/p}$$

و

$$d''(x_1, a_{j1}) = \left((x_1 - a_{j1})^2 + \varepsilon\right)^{1-p/2}$$

و سایر مطالب، همان گونه است که قبلاً برای حالت فاصله خط راست بیان شد.در اینجا نیز به ازای $p = ۲$ روش (۳-۶) را خواهیم داشت.

۳-۳ مساله مکانیابی تک وسیله ای بر روی خط با تابع هدف مینیماکس

در این بخش می خواهیم نقطه جدید $X = (x_1, 0)$ را بر روی خط L به گونه ای مکانیابی کنیم که ماکزیمم فاصله X تا نقاط موجود داده شده a_j، به ازای $j = ۱, ۲, \ldots, n$ مینیمم گردد.

این مساله می تواند کاربردهای زیادی در عرصه های عملی داشته باشد. به طور مثال در مکانیابی خدمات اورژانسی (آمبولانس و آتش نشانی) و در مکانیابی خدمات صنعتی (تحویل فوری) که باید کنار یک خیابان یا جاده احداث گردند و ماکزیمم تاخیر، اهمیت بیشتری نسبت به میانگین یا مجموع تاخیر دارد، از این نوع مکانیابی استفاده می گردد.

شکل کلی مساله به صورت زیر می باشد:

$$\min_{x_1} \max_{j} \quad d\left(X, a_j\right) \qquad j = 1,2,\ldots,n \qquad (۲۲-۳)$$

چون مساله، بهینه سازی تابع هدف در یک بعد می باشد، بنابراین می توان به ازای $j = 1,2,\ldots,n$ ، $\max_{j} d\left(X, a_j\right)$ را به صورت یک تابع $f = \max_{j} d\left(X, a_j\right)$ قرار داد و سپس مقدار تابع هدف مساله :

$$\min_{x_1} f = \max_{j} d\left(X, a_j\right)$$
$$j = 1,\ldots,n$$

را با استفاده از روش های جستجوی خطی برای فاصله های مختلف به دست آورد.

فصل ۴
روش مثلث بزرگ مثلث کوچک

۴-۱ مقدمه

در مسائل مکانیابی گاهی با مسائلی برخورد می کنیم که نامحدب و مشتق ناپذیرند و حتی دارای بهینه های موضعی زیادی هستند. و لذا باید از روش های بهینه سازی سراسری برای حل استفاده کنیم. نوعی روش بهینه سازی سراسری بر مبنای افراز ناحیه و ایجاد شاخه و کران توسط **هرست**[۶۷][۱۹] و **توی** [۶۸] و هرست [۲۰] به ترتیب در سال های ۱۹۸۶و ۱۹۸۸ ارائه شد. روش های **شاخه و کران** [۶۹]، مسائل سخت را با افراز ناحیه شدنی به زیر ناحیه ها (شاخه زدن) و ارزیابی زیر ناحیه ها که شامل محاسبه تابع هدف و یا محدودیت هاست (ایجاد کران) حل می کنند. روش های شاخه و کران از آغاز به طور گسترده در مسائل برنامه ریزی عدد صحیح به کار گرفته شده اند.

یکی از روش های شاخه و کران که برای مسائل پیوسته به کار می رود، روش **مثلث بزرگ مثلث کوچک** [۷۰] است که اصلاح شده روش **مستطیل بزرگ مستطیل کوچک**[۷۱] است که در آن به جای مستطیل از مثلث استفاده شده است [۲۱]. این روش برای مسائل مکانیابی نامحدب و پیوسته پیشنهاد شده است و مزایایی دارد که در ادامه بحث خواهد شد. این الگوریتم دو مزیت دارد، اول اینکه وقتی ناحیه شدنی اجتماعی از چند ضلعی هاست، مثلث ها تمام ناحیه شدنی را می پوشانند و شامل هیچ نقطه نشدنی نمی باشد بنابراین به تست شدنی بودن که کاری وقت گیر است نیازی نداریم. لذا فقط به یک مثلث بندی اولیه نیازمندیم. مزیت دوم اینکه محاسبات کران ها برای خانه ها ساده تر شده اند و حتی در مواردی (وقتی هیچ نقطه تقاضایی در مثلث نباشد) دقیق تر هم شده اند. در مثلث بندی نقاط موجود مکانیابی رئوس مثلث ها در نظر گرفته می شود. برای مثلث بندی ناحیه شدنی از ساختارهای هندسی **گراف ورونوی**[۷۲] و **مثلث بندی دلانی**[۷۳] استفاده می شود که در ادامه این ساختارها را معرفی می کنیم.

۴-۲ گراف ورونوی

گراف ورونوی یک ساختار هندسی است که کاربردهای زیادی در فیزیک، نجوم، رباتیک، زیست شناسی، باستان شناسی و ... دارد [۲۲]. همچنین گراف ورونوی با مثلث بندی دلانی که یکی از مهمترین ساختارهای هندسی است، ارتباط زیادی دارد. از این رو یکی از ابزارهای معروف و پرکاربرد در مسائل مکانیابی گراف ورونوی می باشد.

---

[67] Horst
[68] Tuy
[69] Branch and Bound (B&B)
[70] Big Triangle Small Triangle (BTST)
[71] Big Square Small Square (BSSS)
[72] Voronoi Diagram
[73] Delaunay Triangulation

فرض کنید مجموعه $P=\{p_1,...,p_p\}$ شامل p نقطه در صفحه باشد و هر نقطه از صفحه را به نزدیک ترین نقطه از اعضای P اختصاص می دهیم. بنابراین صفحه به چند ضلعی های $V_1,...V_p$ افراز می شود (شکل ۴-۱) [۱۹]. این افراز را گراف ورونوی، چندضلعی های $V_1,...V_p$ را چندضلعی های ورونوی و P را مجموعه مولد می نامیم. طبق تعریف چند ضلعی ورونوی، اگر نقطه q متعلق به V_i باشد آنگاه p_i نزدیک ترین نقطه مولد به نقطه q است. به همین سبب گاهی V_i را ناحیه یا قلمرو p_i می نامند.

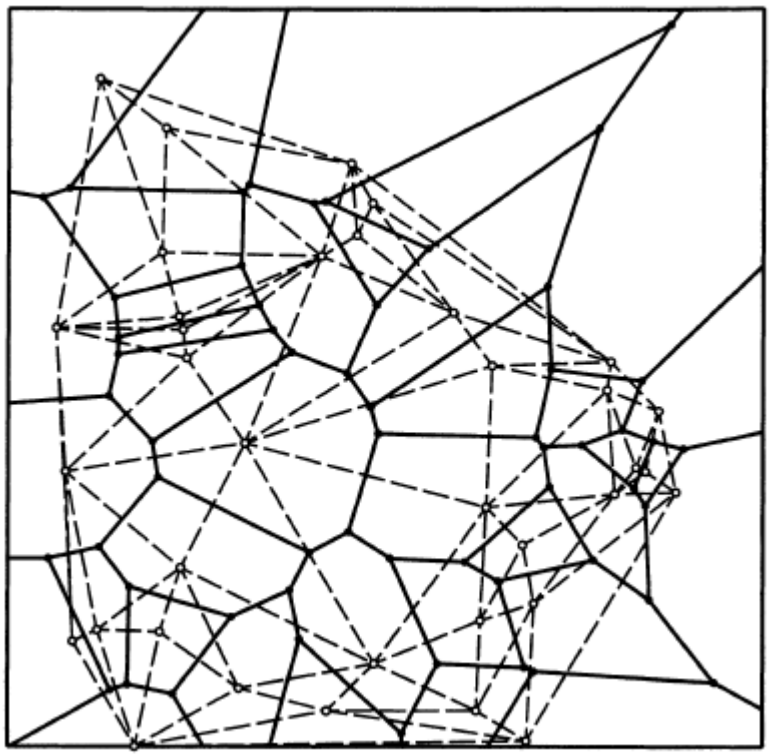
حال تعریف بالا را به صورت ریاضی فرمول بندی می کنیم. فرض کنید $p_i=(x_i,y_i)$ نشان دهنده مکان p_i در فضای R^2 باشد و $\|q-p_i\|$ فاصله اقلیدسی بین p_i و q باشد. چند ضلعی متناظر با p_i را به صورت زیر تعریف می کنیم:

$$V_i=\bigcap_{\substack{j\\ j\neq i}}\left\{q\in R^2 \mid \|q-p_i\|<\|q-p_j\|\right\} \qquad (۱-۴)$$

و همچنین

$$\begin{aligned} w_{ij} &= \partial V_i \bigcap \partial V_j \\ U_{ijk} &= w_{ij}\bigcap w_{jk}\bigcap w_{ki} \end{aligned} \qquad (۲-۴)$$

که w_{ij} (مرز مشترک ناحیه های V_j و V_i) را یال ورونوی و U_{ijk} (نقطه هم راسی حداقل سه یال ورونوی) را یک نقطه ورونوی می نامند.



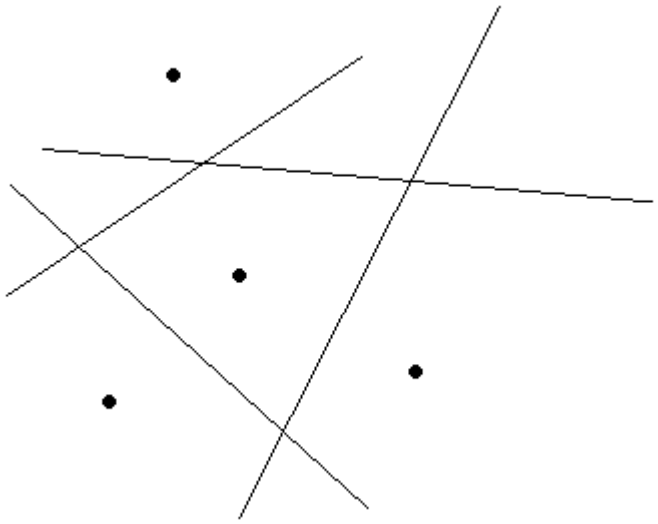
شکل ۴-۱: گراف ورونوی و دوگان آن

برای ساخت گراف ورونوی الگوریتم های زیادی در شاخه هندسه محاسباتی پیشنهاد شده است [۲۴,۲۳]. دو تا از بهترین این روش ها، روش تقسیم و گسترش و روش افزایشی است [۲۵]. پیچیدگی محاسباتی روش اول در بدترین حالت و در حالت میانگین از مرتبه $O(p\log p)$ است. و پیچیدگی محاسباتی روش دوم در بدترین حالت از مرتبه $O(p^2)$ و در حالت میانگین از مرتبه $O(p)$ می باشد.

برای هر دو نقطه p و q در صفحه، عمودمنصف پاره خط $\overline{pq}$، صفحه را به دو نیم صفحه تقسیم می کند. نیم صفحه ای که شامل p است را با $h(p,q)$ و نیم صفحه ای که شامل q است با $h(q,p)$ نشان می دهند. لذا $r \in h(p,q)$ است، اگر و تنها اگر $d(r,p) < d(r,q)$ باشد.

گزاره ۴-۱ ([۲۶]): $V_i = V(p_i) = \bigcap_{\substack{1\le j\le n \\ j\neq i}} h(p_i, p_j)$.

بنا بر گزاره (۴-۱)، $V_i = V(p_i)$ اشتراک $n-1$ نیم صفحه است و از این رو یک ناحیه چند ضلعی محدب و (احتمالا بیکران) با حداکثر $n-1$ یال و $n-1$ راس می باشد. دیگر اینکه گراف ورونوی یک افراز و تقسیم بندی مسطح است که یال هایش مستقیم هستند. بعضی یال ها پاره خط و بعضی دیگر نیم خط می باشند (شکل ۴-۲).



شکل ۴-۲: یال های متناهی و نیمه متناهی گراف ورونوی

قضیه ۴-۱ ([۲۶]): فرض کنید P مجموعه ای از n نقطه در صفحه باشد. اگر این نقاط هم خط باشند آنگاه $V(P)=\bigcup_{1\le i\le n} V_i$ شامل $n-1$ خط موازی است. در غیر این صورت $V(P)$ همبند است و یال هایش پاره خط یا نیم خط هستند.

اثبات: اثبات قسمت اول به سادگی از گزاره (۴-۱) نتیجه می شود. حال فرض می کنیم که تمام نقاط هم خط نباشند. ابتدا نشان می دهیم که یال های $V(P)$ پاره خط یا نیم خط هستند. تا اینجا بنابر اطلاعاتی که داریم، می دانیم یال های $V(P)$ قسمت هایی از خط های مستقیم یعنی عمود منصف های هر جفت از نقاط هستند. به برهان خلف فرض می کنیم که $V(P)$ دارای یک یال مثل e است که خط است. فرض کنیم e روی مرز خانه های ورونوی $V(p_j)$ و $V(p_i)$ و $p_k \in P$ نقطه غیر هم خط با p_j و p_i باشد. عمود منصف $\overline{p_j p_k}$ با e موازی نیست و لذا e را قطع می کند. قسمتی از e که درون $h(p_k, p_j)$ قرار می گیرد، نمی تواند روی مرز $V(p_j)$ باشد، زیرا به نقطه p_k نسبت به p_j نزدیک تر است و این تناقض است. حال ثابت می کنیم که $V(P)$ همبند است. اگر چنین نباشد آنگاه باید چند ضلعی $V(p_i)$ یی موجود باشد که صفحه را به دو قسمت تقسیم کند. چون چند ضلعی های ورونوی محدب هستند، $V(p_i)$ می بایست شامل یک ناحیه باشد که با دو خط موازی کراندار شده است. قبلا ثابت کردیم که یالهای گراف ورونوی نمی توانند خط باشند و این تناقض است. بنابراین حکم بر قرار است.

تعریف۴-۱([۲۷]): گراف مسطح گرافی است که بتوان آن را روی صفحه نشاند، یعنی بتوان آن را طوری رسم کرد که یال هایش همدیگر را قطع نکنند.

لم ۴-۱: (فرمول اویلر)([۲۷]): اگر G گراف همبند و مسطح روی صفحه باشد و e تعداد یال های آن، v تعداد راس های G، و f تعداد وجه های G باشد، آنگاه:

$$v-e+f=2$$

قضیه ۴-۲ ([۲۶]): برای $n\ge 3$، تعداد رئوس گراف ورونوی مربوط به n نقطه در صفحه، حداکثر $2n-5$ و تعداد یال ها حداکثر $3n-6$است.

اثبات: اگر نقاط هم خط باشند، نتیجه از قضیه (۴-۱) بدست می آید. این قضیه را با استفاده از فرمول اویلر و با فرض هم خط نبودن نقاط اثبات می کنیم. بنابر فرمول اویلر، برای هر گراف همبند مسطح که بر روی صفحه نشانده شده است و در آن m_v بیانگر تعداد گره ها، m_e تعداد یال ها و m_f تعداد وجه ها باشند رابطه زیر بر قرار است:

$$m_v-m_e+m_f=2 \qquad (۴-۳)$$

اما چون $V(p)$ شامل یال های نیمه نامتناهی است نمی توانیم از فرمول اویلر برای آن استفاده کنیم. به این منظور یک راس فرضی به نام v_∞ را در بینهایت در نظر می گیریم و به مجموعه رئوس اضافه می کنیم سپس تمام یال های نیمه نا متناهی را به آن متصل می کنیم. حال یک گراف مسطح همبند داریم و می توانیم از فرمول اویلر استفاده کنیم (شکل۴-۳).

رابطه زیر بین n_v تعداد رئوس گراف ورونوی، n_e تعداد یال های $V(p)$ و n تعداد سرویس دهنده ها برقرار است:

$$(n_v+1)-n_e+n=2 \qquad (۴-۴)$$

علاوه بر این هر یال در گراف مورد بحث دارای دقیقا دو راس است. بنابراین اگر درجه همه رئوس را با هم جمع کنیم برابر خواهد شد با دو برابر تعداد یال ها. چون هر راس (به انضمام راس v_∞)، حداقل از درجه ۳ است، لذا داریم:

$$2n_e \geq 3(n_v+1) \qquad (۴-۵)$$

لذا با استفاده از رابطه (۴-۵) داریم:

$$2(n_v+1)+2n-4 \geq 3(n_v+1)$$

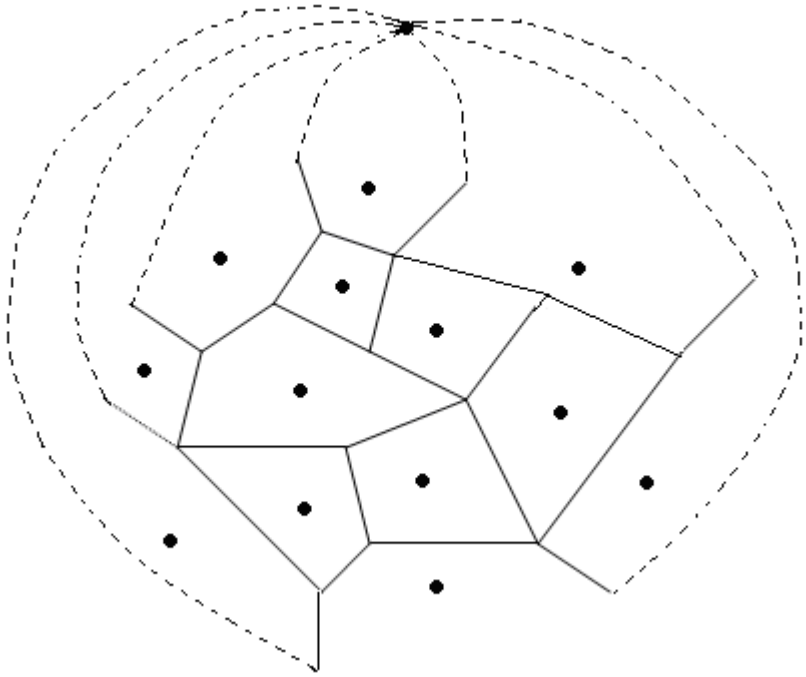
بنابراین:

$$2n_e \geq 3(2-n-n_e) \quad \text{و} \quad n_v \leq 2n-5$$

پس:

$$n_e \leq 3n-6$$

لذا حکم قضیه حاصل می شود.



شکل۴-۳: گراف همبند و مسطح شده گراف ورونوی با یالهای نیمه متناهی و راس بینهایت

قضیه ۴-۳ ([۲۶]): نتایج زیر برای گراف ورونوی $V(P)$ از مجموعه نقاط P برقرارند:

i) نقطه q یک راس از $V(P)$ است اگر و تنها اگر حداقل سه نقطه روی محیط بزرگترین دایره خالی $C_p(q)$ درون $V(P)$، قرار گیرند.

j) عمود منصف p_j و p_i یک یال $V(P)$ است، اگر و تنها اگر نقطه ای مانند q روی عمود منصف موجود باشد که هیچ نقطه دیگری از P غیر از p_j و p_i روی محیط $C_p(q)$ آن قرار نگیرد.

اثبات: (i) : فرض می کنیم که q نقطه ای باشد که $C_p(q)$ شامل حداقل سه عضو از P روی محیط اش باشد. این نقاط را p_i, p_j, p_k در نظر می گیریم. از آنجایی که درون $C_p(q)$ خالی است، q باید روی مرز $V(p_i), V(p_j), V(p_k)$ قرار گیرد و لذا یک راس $V(P)$ است.

برای اثبات عکس، هر راس q از $V(P)$ با حداقل سه یال مجاور است یعنی با حداقل سه خانه گراف ورونوی، $V(p_i), V(p_j), V(p_k)$ مجاور می باشد. راس q باید از سه نقطه p_i, p_j, p_k به یک فاصله باشد، زیرا در غیر این صورت p_i, p_j, p_k در q مشترک نخواهند شد. بنابراین دایره ای که p_i, p_j, p_k روی محیط آن قرار می گیرند، شامل هیچ عضو دیگری از P در درون خود نیست.

(ii) فرض می کنیم که q دارای ویژگی های بیان شده در قضیه باشد. چون $C_p(q)$ شامل هیچ عضوی از P در درونش نیست و p_j و p_i روی محیط آن قرار دارند، به ازای $1 \le k \le n$ داریم: $d(q, p_i) = d(q, p_j) \le d(p_i, p_k)$. لذا نتیجه می شود که q نمی تواند یک راس باشد. پس q روی یک یال از $V(P)$ قرار دارد که همان عمود منصف p_j و p_i می باشد.

برای عکس، فرض می کنیم که عمود منصف p_j و p_i یک یال ورونوی را تشکیل دهد. بزرگترین دایره خالی از هر نقطه ای مثل q در درون این یال باید شامل p_j و p_i روی محیط خود باشد و هیچ عضو دیگری از P روی آن قرار نمی گیرد.

۴-۳ مثلث بندی دلانی

فرض کنیم $P=\{p_1,...,p_2\}$ مجموعه ای از نقاط روی صفحه باشد. افراز S را افراز مسطح ماکزیمال می گویند هرگاه با افزودن یک یال به S گراف حاصل مسطح نباشد. یک مثلث بندی از P را یک افراز مسطح ماکزیمال که رئوس آن از P باشند، تعریف می کنیم. دوگان گراف ورونوی را با G نشان می دهیم. دوگان گراف G گرافی است که متناظر با هر وجه از G یک راس و متناظر با هر دو وجه مجاور در G، یک یال دارد. اگر گراف G را با یالهای مستقیم (پاره خط های راست و بدون انحنا) در نظر بگیریم، گراف حاصل را گراف دلانی نامیده و با $DG(P)$ نشان می دهیم. نقاط و یال های مثلث بندی دلانی متناظر با مولد های گراف ورونوی و پاره خط های واصل مولد های همسایه است. در حقیقت یک تناظر یک به یک بین رئوس $V(P)$ و وجوه کراندار G وجود دارد. در شکل۴-۱ نمودار ورونوی (خطوط ضخیم) و مثلث بندی دلانی (خط چین ها) تولید شده توسط نقاط تصادفی در مستطیل واحد را مشاهده می کنید.

۴-۳-۱ الگوریتم مثلث بزرگ مثلث کوچک

الگوریتمی که در ادامه می آید را می توان برای نواحی که اجتماعی از چند ضلعی های محدب هستند به کار برد. هر ناحیه ای را می توان با دقت دلخواه به وسیله اجتماعی از چند ضلعی های محدب تقریب زد. ممکن است بعضی از نقاط تقاضا بیرون ناحیه شدنی باشد که از آنها صرف نظر می شود.

۱- مرحله مثلث بندی (فاز۱)

ناحیه شدنی اجتماعی از چند ضلعی های محدب است. هر چند ضلعی را با استفاده از رئوسش و تقاضایی که درون آن قرار می گیرند مثلث بندی می کنیم. نقاط تقاضایی که درون چند ضلعی ها قرار ندارند را در مثلث بندی استفاده نمی کنیم.

۲- مرحله ارزیابی (فاز۲)

مقدار تابع هدف را در مرکز (مرکز ثقل سه راس با وزن های برابر) هر مثلث محاسبه می شود. کمترین مقدار تابع هدف در مراکز مثلث ها را به عنوان کران بالای اولیه UB منظور می کنیم.

۳- یک کران پایین برای تابع هدف در هر مثلث LB محاسبه می شود.

۴- از تمام مثلث هایی که کران پایین شان بزرگتراز $\frac{UB}{1+\varepsilon}$ باشد صرف نظر می کنیم. مثلث های باقیمانده مجموعه اولیه مثلث ها T_0 را تشکیل می دهند.

۵- مرحله شاخه و کران (فاز۳)

کوچکترین کران پایین را پیدا می کنیم ($LB_{\min}$). اگر $LB_{\min} \geq \frac{UB}{1+\varepsilon}$، الگوریتم متوقف می شود، UB جواب است.

۶- مثلثی که کوچکترین کران پایین را دارد، $LB_{\min}$، به چهار مثلث مجزا افزار می شود. رئوس این چهار مثلث شامل رئوس مثلث اصلی و مرکز اضلاع آن است. این چهار مثلث همگی متشابه با مثلث اصلی هستند و اضلاعشان نصف اضلاع متناظر در مثلث اصلی است.

۷- یکی از مثلث ها جایگزین مثلث اصلی شده و سه مثلث دیگر به لیست مثلث ها اضافه می شوند.

۸- مقدار تابع هدف در مرکز هر یک از چهار مثلث کوچک محاسبه می شود و کران بالا در صورت نیاز بروز می شود.

۹- یک کران پایین برای هر مثلث کوچک محاسبه شود.

۱۰- تمام مثلث های درون مجموعه T که کران پایین شان بزرگتر از $\frac{UB}{1+\varepsilon}$ است را کنار می گذاریم. (اگر UB تغییر کرد، تمام مثلث ها باید مورد بررسی قرار گیرند) به مرحله ۵ باز گرد.

نکاتی در مورد الگوریتم که باید به آنها توجه کنیم:

۱- مثلث بندی هر چند ضلعی، مجموعه ای از مثلث ها را به دست می دهد.

۲- اجتماع همه مجموعه های مثلث ها، T، افرازی از ناحیه شدنی به مثلث را تشکیل می دهد که هیچ نقطه تقاضایی در درون مثلث ها وجود ندارد. بعضی از رئوس و اضلاع ممکن است به چند مثلث تعلق داشته باشند.

۳- در مثلث بندی ممکن است بعضی از مثلث ها دراز و باریک باشند. کران ها در درون چنین مثلث هایی ممکن است دارای دقت کافی نبا شند در این حالت می توان مثلث های دراز و باریک را (یک زاویه بزرگتر از ۱۲۰درجه دارند) به چهار مثلث کوچکتر تقسیم کرد. برای این منظور بلند ترین ضلع را به چهار قسمت مساوی تقسیم کرده و از راس مقابل آن به این نقاط وصل می کنیم.

۴- بیشترین تعداد مثلث های ایجاد شده در طول فرآیند شاخه و کران را با $T_{\max}$ نشان می دهیم.

۵- چند ضلعی های نا محدب را به راحتی با اجتماعی از چند ضلعی های محدب می توان به دست آورد و لذا به سادگی می توان آن ها را مثلث بندی کرد.

فصل ۵
مساله زمانبندی و مکانیابی توام مدل تک ماشینی روی صفحه

۵-۱ مقدمه

تئوری زمانبندی و مکانیابی دو زمینه مهم در تحقیق در عملیات هستند، که کاربردهای زیادی دارند. ما در این فصل می خواهیم به بررسی مساله توام زمانبندی و مکانیابی بپردازیم. به عنوان مثال کانتینرهایی در ساحل قرار دارند و باید توسط کشتی ها بارگیری شوند. در این کاربرد می خواهیم بهترین مکان برای کشتی ها در لنگرگاه پیدا کنیم (مکانیابی) و چگونگی فرایند بارگیری کانتینرها را مشخص کنیم (زمانبندی). بطور معمول هر کسی ابتدا مساله مکانیابی را حل می کند و سپس مساله زمانبندی را حل خواهد کرد.

وقتی که مساله با نگرش کلی مدل بندی می شود، بهترین مکان برای کشتی ها و ترتیب بارگیری کانتینرها تواما بدست می آید. هرچند که جواب بهینه ممکن است شامل جواب بهینه برای دو مساله به صورت مجزا نباشد. یکی دیگر از کاربردهای این مساله، استفاده از ماشین های متحرک در برنامه ریزی تولید به ازای هر محصول است. که شامل مجموعه ای از کارها بوده و ماشین ها می توانند مکان پردازششان را تغییر دهند. واضح است که برنامه ریزی توام مکانیابی ماشین ها و زمانبندی خط تولید متناظر آن، منطقی می باشد. بدیهی است که در نظر گرفتن سیستم های واقعی بسیار پیچیده است و حل آن مشکل می باشد. بنابراین مدل ها و شیوه های حل برای بعضی حالات ویژه، مثل حالات تک ماشینی می توانند به عنوان مقیاس هایی برای حل سیستم های پیچیده تر ، مثلا برای حالت های چند ماشینی استفاده گردند.

مساله توام زمانبندی- مکانیابی(ScheLoc)[۷۴] اولین بار توسط هاماچر[۷۵] و هنز[۷۶][۲۸] مطرح شد. سپس مقاله ای توسط هاماچر و هنز منتشر شد که در آن ماشین ها می توانند در هر جای شبکه قرار گیرند (N-ScheLoc) [۲۹]. بررسی جزئیات بیشتر روی این مساله توسط هنز [۳۰] صورت گرفت. اولین نتیجه در زمینه مساله مکانیابی- زمانبندی بر روی صفحه (P-ScheLoc) که ماشین ها می توانند هر جای یک صفحه قرار گیرند توسط الویکیز[۷۷] و همکاران[۳۲,۳۱] پایه گذاری شد. آنها برای اولین بار مساله کل زمان اتمام کار[۷۸] مدل تک ماشینی P-ScheLoc را بیان کردند و سه الگوریتم با پیچیدگی زمانی چند جمله ای برای آن ارائه دادند. اولین الگوریتم مجموعه ای متناهی از جواب های مناسب را ارائه می دهد، در حالی که الگوریتم های دوم و سوم از تکنیک برنامه ریزی خطی استفاده می کنند. رویه دو الگوریتم اول بهینه سازی کلی است، در حالی که

---

[74] Scheduling and location problems simultaneously
[75] Hamacher
[76] Hennes
[77] Elvikis
[78] Makespan

الگوریتم سوم به روش جستجوی محلی[۷۹] و الگوریتمی ابتکاری[۸۰] است. مساله و الگوریتمی که برای مساله زمانبندی و مکانیابی توام تک ماشینی بر روی صفحه در این پایان نامه مورد بررسی قرار می گیرد توسط درزنر[۸۱] و کالش[۸۲] [۳۳] ارائه شده است.

۲-۵ مساله توام زمانبندی مکانیابی تک ماشینی بر روی صفحه

فرض کنید $\varphi = \{1,2,...,n\}$ مجموعه ای از n کار با زمان پردازش نامنفی p_i ، $i \in \varphi$، باشد که باید به صورت غیر قابل قطع روی تک ماشین داده شده M زمانبندی شوند، به طوری که ماشین M می تواند هر جای صفحه R^2 قرار گیرد و هر کار $i \in \varphi$ دارای مکان ذخیره سازی[۸۳] $a_i \in R^2$ می باشد.

از این رو مساله زمانبندی-مکانیابی تک ماشینی برروی صفحه (1-P-ScheLoc) انتخاب مکان ماشین $x \in R^2$ است، تحت شرایطی که مجموعه کارهای φ کاملا پردازش شوند و همه شرایط پردازش، برقرار باشد. هدف ما بهینه کردن بعضی تابع هدف های زمانبندی است که نه تنها به دنباله کارها بستگی دارد، بلکه به انتخاب x نیز وابسته است.

در مساله 1-P-ScheLoc، هر کار $i \in \varphi$ توسط پارامترهای زیر مشخص می شود. زمان فرارسی ذخیره[۸۴] $\sigma_i \geq o$، که در لجستیک های کانتینری، σ_i می تواند به صورت زمانی که یک وسیله (ترابر) می تواند یک کانتینر (کار) را از مکان ذخیره سازی اش a_i، به کشتی (ماشین) M انتقال دهد، تفسیر شود. اگر $\sigma_i = 0$ باشد، آنگاه کار i در شروع پردازش دنباله، تقریبا در مکان ذخیره سازی اش قابل دسترس است. یعنی انتقال کار i از a_i به M می تواند در زمان صفر شروع گردد.

چون σ_i مثبت است بنابراین می تواند در طی پردازش کارهای دیگر حرکت داده شود. سرعت انتقال $v_i > 0$، سرعت حرکت کار i می باشد یا معادل است با سرعت تغییر مکان. زمانی که کار i می تواند پردازشش را شروع کند، به سرعت ورودش به ماشین M نیز بستگی دارد. بنابراین این زمان را به عنوان زمان آماده سازی کار i تفسیر می کنند. فرض کنید d_i یک تابع فاصله کلی روی R^2 باشد، $\tau_i := \frac{1}{v_i} > o$، آنگاه متناظر با a_i، $r_i(x) = \sigma_i + \tau_i d_i(a_i, x)$ متغیر زمان آماده سازی کار i برای ماشین M نامیده می شود که بستگی به مکان ماشین، $x \in R^2$ نیز دارد. دنباله کارها که روی ماشین پردازش می شوند، با یک جایگشت π از $\{1,...,n\}$ تعریف می شود، که $\pi(j) = i$ به این معنی است که کار i، jامین کاری است که پردازش می شود. مجموعه تمام جایگشت های

---

[79] Local search
[80] Heuristic
[81] Drezner
[82] Marcel T. Kalsch
[83] Storage location
[84] Storage arrival time

$\{1,...,n\}$ با Π_n نشان داده می شود. به ازای هر دنباله $\pi \in \Pi_n$ و هر مکان $x \in R^2$ برای ماشین، زمان کامل شدن به ازای هر کار $i \in \varphi$ با استفاده از رابطه بازگشتی زیر تعریف می شود:

$$C_{\pi(1)}(x) = r_{\pi(1)}(x) + p_{\pi(1)} \qquad (۱-۵)$$

$$C_{\pi(j)}(x) = \max\left\{C_{\pi(j-1)}(x), r_{\pi(j)}(x)\right\} + p_{\pi(j)} \qquad j \in \{2,3,...,n\} \qquad (۲-۵)$$

که $p_{\pi(j)}$، زمان پردازش کار $\pi(j)$ تعریف می شود. توجه کنید که زمان کامل شدن در فرمول های بالا می تواند صریحا به صورت زیر نیز بیان گردد.

$$C_{\pi(j)}(x) = \max\left\{r_{\pi(j)}(x) + p_{\pi(j)}, r_{\pi(j-1)}(x) + \sum_{s=j-1}^{j} p_{\pi(s)}, ..., r_{\pi(1)}(x) + \sum_{s=1}^{j} p_{\pi(s)}\right\} \qquad (۳-۵)$$

به ازای هر $j \in \{1,2,...,n\}$.

در نهایت ماکزیمم زمان کامل شدن (کل زمان اتمام کار) در $x \in R^2$ و برای یک $\pi \in \Pi_n$ داده شده برابر است با:

$$\max\{C_1(x),...,C_n(x)\} = C_{\pi(n)}(x) \qquad (۴-۵)$$

و مجموع زمان های کامل شدن در $x \in R^2$ و برای $\pi \in \Pi_n$ برابر است با:

$$\sum_{i=1}^{n} C_i(x) = \sum_{j=1}^{n} C_{\pi(j)}(x) \qquad (۵-۵)$$

برای مشخص شدن مدل و ارزیابی جواب مساله زمانبندی-مکانیابی دو مساله خاص را بررسی می کنیم. یکی مساله کل زمان اتمام کار 1-P-ScheLoc و دیگری مساله مجموع زمان های کامل شدن کارها در 1-P-ScheLoc یا (1-P-ScheLocTCP).

۵-۲-۱ مساله کل زمان اتمام کار 1-P-ScheLoc

در این بخش روی مساله کل زمان اتمام کار 1-P-ScheLoc تمرکز می کنیم، به قسمی که $C_{\pi(n)}(x)$ مینیمم گردد. با استفاده از روابط (۵-۳) و (۵-۴) این مساله را می توان به صورت زیر نوشت:

$$\min C_{\pi(n)}(x) = \max\left\{ r_{\pi(n)}(x) + p_{\pi(n)}, r_{\pi(n-1)}(x) + \sum_{s=n-1}^{n} p_{\pi(s)}, \ldots, r_{\pi(1)}(x) + \sum_{s=1}^{n} p_{\pi(s)} \right\}$$

$$s.t \quad x \in R^2 \qquad (۶-۵)$$
$$\pi \in \Pi_n$$

از این رو $C_{\pi(n)}(x)$، تابع هدف کل زمان اتمام کار، به x و π وابسته است.

اگر به ازای تمام $i \in \varphi$ قرار دهیم $p_i = 0$ ، $\sigma_i = 0$ و $\tau_i = 0$ ، آنگاه کل زمان اتمام کار مساله 1-P-ScheLoc به مساله مکانیابی ساده ۱- مرکز[۸۵] کلاسیک تقلیل می یابد. اگر x را ثابت فرض کنیم، آنگاه باید یک مساله کلاسیک زمانبندی کل زمان اتمام کار با زمان های آماده سازی ثابت یا $(1/r_j / C_{\max})$ را حل کنیم. بعلاوه اگر دنباله π داده شده باشد، آنگاه باید یک مساله مکانیابی تک وسیله ای را برای بدست آوردن مکان بهینه ماشین حل کنیم. واضح است که برای توابع فاصله محدب و یک دنباله ثابت $\pi \in \Pi_n$ ، تابع هدف $C_{\pi(n)}(x)$ روی R^2 محدب است.

یادآوری می کنیم که به ازای یک مکان داده شده ماشین، $x \in R^2$، مساله کل زمان اتمام کار 1-P-ScheLoc، به یک مساله زمانبندی کل زمان اتمام کار کلاسیک با زمان های آماده سازی ثابت $r_i(x) = r_i$ به ازای $i \in \varphi$ تقلیل می یابد. در این حالت برای بدست آوردن دنباله بهینه برای کارها می توانیم از قانون زودترین زمان آماده سازی (ERD) استفاده کنیم، که کارها در ترتیب غیر نزولی زمان آماده سازی مرتب می شوند ([۳۴، ۳۵، ۳۶] را ببینید). بنابراین به ازای هر مکان ماشین $X \in R^2$ ، می توان یک دنباله بهینه موضعی از کارها با استفاده از قانون ScheLoc ERD بدست آورد. یعنی مرتب سازی کارهای $i \in \varphi$ در ترتیب غیر نزولی از زمان آماده سازی شان:

$$r_{\pi(1)}(x) \le r_{\pi(2)}(x) \le \ldots \le r_{\pi(n)}(x) \qquad (۷-۵)$$

بنابراین متغیرهای $\pi \in \Pi_n$ مساله کل زمان اتمام کار 1-P-ScheLoc، تنها به ترتیب غیر نزولی زمان های آماده سازی و زمان های آماده سازی تنها به متغیر $x \in R^2$ وابسته است. با استفاده از این نتایج به فرمول بندی جدیدی برای مساله، به صورت زیر می رسیم:

$$\min C_{\pi(n)}(x) = \max\left\{ r_{\pi(n)}(x) + p_{\pi(n)}, r_{\pi(n-1)}(x) + \sum_{s=n-1}^{n} p_{\pi(s)}, \ldots, r_{\pi(1)}(x) + \sum_{s=1}^{n} p_{\pi(s)} \right\}$$

$$s.t \quad r_{\pi(1)} \le \ldots \le r_{\pi(n)}$$
$$x \in R^2$$
$$\pi \in \Pi_n$$

[85] 1-center

در ادامه سه شرط کافی، برای حالتی که یکی از مکان های کارها، مکان بهینه ScheLoc برای ماشین است، توصیف می کنیم. نتایجی که یک کران پایین بدیهی به صورت $LB_{\max} := \min_{i\in\varphi}\{\sigma_i\}+\sum_{i\in\varphi} p_i$ که در تمام حالت ها بدست می آید، نتیجه می دهد.

گزاره ۵-۱ ([۳۷]): فرض کنید $i \in \arg\min\{\sigma_s : s=1,...,n\}$ و $x = a_i$ و فرض کنید $\pi_{a_i} = (\pi(1)=i,\pi(2),...,\pi(n))$ یک دنباله محاسبه شده با قانون ScheLoc ERD با استفاده از $r_s(a_i)$ که $s \in \varphi$، باشد. آنگاه دنباله $\pi^* = \pi_{a_i}$ و مکان ماشین $x_{\pi^*} = a_i$ یک جواب بهینه برای مساله کل زمان اتمام کار 1-P-ScheLoc تعیین می کند، اگر حداقل در یکی از سه شرط زیر صدق کند.

$a)\ \sigma_i + p_i \geq r_k(a_i) = \sigma_k + \tau_k d_k(a_k, a_i) \qquad \forall k \in \{1,...,n\}$

$b)\ \sigma_i + \sum_{j=1,...,s} p_{\pi(j)} \geq r_{\pi(s+1)}(a_i) \qquad \forall s \in \{1,...,n-1\}$

$c)\ LB_{\max} = C_{\pi(n)}(a_i)$

در گزاره بالا اثبات [۳۷] می شود که اگر شرط (a) برقرار باشد آنگاه شروط (b) و (c) نیز برقرار است. همچنین اگر شرط (b) برقرار باشد، آنگاه شرط (c) برقرار است. یعنی $a \Rightarrow b \Rightarrow c$.

کران پایین دوم LB_{cent} برای مساله کل زمان اتمام کار 1-P-ScheLoc، از حل مساله مکانیابی ۱-مرکز زیر بدست می آید.

$$\min_{X\in R^2} \max_{i\in\varphi}\{\sigma_i + \tau_i d_i(a_i, x) + p_i\} \qquad (۸-۵)$$

یک روش حل برای این مساله استفاده از نرم اقلیدسی است که در مقاله درزنر آمده است[۳۸]. مساله با استفاده از نرم های کلی توسط میچلوت[۸۶] و پلاستریا[۸۷] [۳۹] بحث شده است.

فرض کنید x_{cent} نقطه بهینه وزن دار در مساله (۵-۸) باشد. و فرض کنید $LB_{cent} = \max_{i\in\varphi}\{r_i(X_{cent}) + p_i\}$ یک کران پایین برای مساله کل زمان اتمام کار 1-P-ScheLoc باشد. چون به ازای هر $x \in R^2$ و هر $\pi \in \Pi_n$ داریم :

$$C_{\pi(n)}(x) \geq \max_{i\in\varphi}\{r_i(x) + p_i\} \geq LB_{cent}$$

از این رو معیار بهینگی کافی دیگری به صورت زیر بدست می آوریم.

گزاره ۵-۲([۳۷]): فرض کنید x_{cent} نقطه بهینه وزن دار مساله ۱-مرکز باشد و $\pi_{x_{cent}}$ دنباله ای محاسبه شده با قانون ScheLoc ERD با استفاده از $r_s(X_{cent})$ که $s \in \varphi$، باشد. آنگاه دنباله

---

[86] Michelot
[87] Plastria

$\pi^* = \pi_{x_{cent}}$ و مکان ماشین $x_{\pi^*} = x_{cent}$، یک جواب بهینه برای مساله کل زمان اتمام کار 1-P-ScheLoc تعیین می کند، اگر $LB_{cent} = C_{\pi(n)}(x_{cent})$.

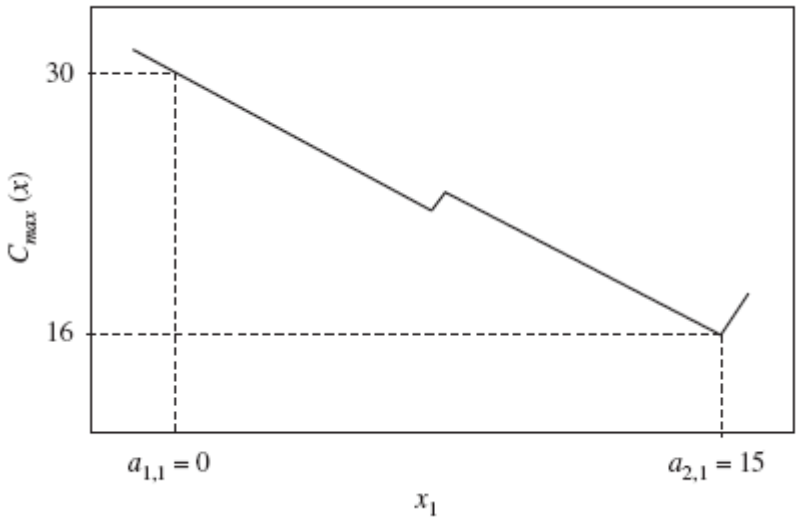
حال اگر هیچ یک از شرایط بهینگی پیشین برقرار نباشد. آنگاه باید الگوریتمی مفیدی برای حل مساله ScheLoc ارائه گردد که در ادامه آمده است.

باید توجه داشته باشیم که مساله کل زمان اتمام کار 1-P-ScheLoc یک مساله بهینه سازی کلی است. چون تابع هدف $C_{\max}(x) = \min_{\pi \in \Pi_n} \{C_{\pi(n)}(x)\}$ در حالت کلی روی R^2 نامحدب است. (مثال ۵-۱ را ببینید).

مثال ۵-۱ ([۳۷]): دو کار را با مکان های ذخیره سازی $a_1 = (0,0)$ و $a_2 = (15,0)$ و با نرم فاصله $d_i = \ell_q \quad (1 \le q \le \infty)$ ، برای i=1,2 در نظر بگیرید. فرض کنید $p_1 = 1, p_2 = 15$ ، $v_1 = v_2 = 1$, $\sigma_1 = \sigma_2 = 0$ و $\Pi_2 = \{(1,2),(2,1)\}$ باشد:

$$C_{\max}(a_1) = \min_{\pi \in \Pi_2} \{C_{\pi(2)}(a_1)\} = \max\{r_2(a_1), r_1(a_1) + p_1\} + p_2 = \max\{15, 0+1\} + 15 = 30$$
$$C_{\max}(a_2) = \min_{\pi \in \Pi_2} \{C_{\pi(2)}(a_2)\} = \max\{r_1(a_2), r_2(a_2) + p_2\} + p_1 = \max\{15, 0+15\} + 1 = 16$$
$$C_{\max}(0.5 \times (a_1 + a_2)) = \max\{7.5, 7.5+1\} + 15 = \max\{7.5, 7.5+15\} + 1$$
$$= 23.5 > 0.5 \times (30 + 16 = 23)$$

بنابراین $C_{\max}(x)$ در حالت کلی روی R^2 نامحدب است (شکل ۵-۱([۳۳]) را ببینید).



شکل ۵-۱ شرح $C_{\max}(X)$ مثال ۵-۱

۵-۲-۲ مساله مجموع زمان های کامل شدن 1-P-ScheLoc

در این بخش مساله مجموع زمان های کامل شدن 1-P-ScheLoc با زمان های پردازش یکسان، یعنی به ازای هر $i \in \varphi$ ، $p_i = p$ را مورد بررسی قرار می دهیم. با استفاده از روابط (۵-۱) و (۵-۲) و (۵-۵) این مساله را می توان به صورت زیر نوشت:

$$\begin{aligned} \min \quad & \sum_{j=1}^{n} C_{\pi(j)}(x) \\ s.t \quad & C_{\pi(1)}(x) = r_{\pi(1)}(x) + p \\ & C_{\pi(j)}(x) = \max\left\{C_{\pi(j-1)}(x), r_{\pi(j)}(x)\right\} + p \qquad \forall j \in \{2,\ldots,n\} \qquad (۹-۵) \\ & x \in R^2 \\ & \pi \in \Pi_n \end{aligned}$$

واضح است که تابع هدف مساله مجموع زمان های کامل شدن نیز به متغیرهای π و x وابسته است. واضح است که اگر به ازای هر $i \in \varphi$ ، $\tau_i = 1, \sigma_i = 0, p = 0$ باشد، آنگاه مساله 1-P-ScheLoc TCP به یک مساله مکانیابی ۱- میانه کلاسیک تقلیل می یابد. اگر x را ثابت فرض کنیم، آنگاه باید یک مساله زمانبندی مجموع زمان های کامل شدن با زمان های آماده سازی ثابت و زمان های پردازش برابر یا ($1/r_i, p_i = p/\sum C_i$) را حل کنیم. علاوه بر این برای دنباله داده شده π ، باید یک مساله مکانیابی تک وسیله ای را برای بدست آوردن مکان بهینه ماشین حل کنیم. بدیهی است که برای توابع فاصله محدب و دنباله ثابت $\pi \in \Pi_n$ ، تابع هدف $\sum_{j=1}^{n} C_{\pi(j)}(x)$ نیز روی R^2 محدب است.

یادآوری می کنیم که مساله 1-P-ScheLoc TCP ، به ازای یک مکان $x \in R^2$ برای ماشین، به مساله زمانبندی مجموع زمان های کامل شدن با زمان های آماده سازی ثابت $r_i(x) = r_i$ ، $i \in \varphi$ ، و زمان های پردازش برابر ($1/r_i, p_i = p/\sum C_i$) تقلیل می یابد. این مساله نیز می تواند با استفاده از مرتب کردن زمان های آماده سازی در تر تیب غیر نزولی، حل گردد. یعنی می توان قانون ERD را برای بدست آوردن یک دنباله بهینه به کار برد.([۳۵، ۴۰] را ببینید). از این رو به ازای هر مکان ماشین $x \in R^2$ می توان قانون ScheLoc ERD را برای بدست آوردن یک دنباله کار بهینه موضعی به کار برد. با استفاده این نتایج به یک فرمول بندی جدید برای مساله به صورت زیر می رسیم:

$$
\begin{aligned}
\min \quad & \sum_{j=1}^{n} C_{\pi(j)}(x) \\
s.t \quad & C_{\pi(j)}(x) \geq C_{\pi(j-1)}(x) + p \qquad \forall j \in \{2,\ldots,n\} \\
& C_{\pi(j)}(x) \geq r_{\pi(j)}(x) + p \qquad \forall j \in \{1,\ldots,n\} \qquad (۱۰-۵) \\
& r_{\pi(1)}(x) \leq \ldots \leq r_{\pi(n)}(x) \\
& x \in R^2 \\
& \pi \in \Pi_n
\end{aligned}
$$

واضح است که محدودیت هـای دوم و سـوم رابطـه هـای (۵-۹) و (۵-۱۰) بـه متغیرهـای π و x وابسته اند. توجه کنید که محدودیت سوم رابطه (۵-۱۰) می تواند بـا $C_i(x) \geq r_i(x) + p$ بـه ازای هر $i \in \varphi$، تعویض گردد. در ادامه سه شرط کافی برای حالتی که یکی از مکان های کارهـا، مکـان بهینه ScheLoc برای ماشین است ارائه می شود. . این شرایط یک کران پایین بـدیهی بـه صـورت $LB_{sum} := \sum_{j=1}^{n} (\min_{i\in\varphi} \{\sigma_i\} + \sum_{t=1}^{j} p)$ که در تمام حالت ها بدست می آید را نتیجه می دهد. توجه کنید که LB_{sum} می تواند از محدودیت دوم رابطه (۵-۱۰) بدست آید.

گـــــــــزاره ۵-۳([۳۳]): فـــــــرض کنیـــــــد $i \in \arg\min\{\sigma_s : s = 1,\ldots,n\}$ و $x = a_i$ و $\pi_{a_i} = (\pi(1) = i, \pi(2),\ldots,\pi(n))$ یک دنباله محاسبه شده با قانون ScheLoc ERD با اسـتفاده از $r_s(a_i)$ که $s \in \varphi$، باشد. آنگاه دنباله $\pi^* = \pi_{a_i}$ و مکان ماشین $x_{\pi^*} = a_i$ یک جواب بهینـه بـرای مساله 1-P-ScheLoc TCP با زمان های پردازش برابر تعیین می کند اگر حـداقل در یکـی از سـه شرط زیر صدق کند.

$a)\ C_{\pi(1)}(a_i) = \sigma_i + p \geq r_k(a_i) \qquad \forall k \in \{1,\ldots,n\}$

$b)\ C_{\pi(s)} \geq r_{\pi(s+1)}(a_i) \qquad \forall s \in \{1,\ldots,n-1\}$

$c)\ LB_{sum} = \sum_{j=1}^{n} C_{\pi(j)}(a_i)$

در گزاره (۵-۳) اثبات می شود [۳۳] که اگر شرط (a) برقرار باشد، آنگـاه شـروط (b) و (c) نیـز برقرار می باشد. و اگر شرط (b) برقرار باشد، آنگاه شرط (c) نیز برقرار است. یعنی $a \Rightarrow b \Rightarrow c$.

دومین کران پایین برای مساله 1-P-ScheLoc TCP می تواند بـا اسـتفاده از اولـین محـدودیت رابطه (۵-۱۰) بدست آید. از این رو یک کران پایین جدید LB_{med}، با حل مساله مکانیابی ۱-میانـه وزن دار زیر بدست می آید.

$$\min_{X\in R^2} \sum_{i\in\varphi} (\sigma_i + \tau_i d_i(a_i, x) + p) \qquad (۱۱-۵)$$

مساله مکانیابی ۱-میانه (مساله کمترین مجموع مکانیابی تک وسیله ای) بر روی صفحه به عنوان مساله وبر[۸۸] شناخته شده است. که در فصل سوم به طور مفصل بحث شده است.

فرض کنید x_{med} یک نقطه بهینه مساله ۱-میانه وزن دار (۵-۱۱) باشد. و فرض کنید $LB_{med} := \sum_{i\in\varphi}(r_i(x_{med})+p)$ یک کران پایین برای مساله 1-P-ScheLoc TCP با زمان های پردازش برابر باشد. در زیر شرایط کافی برای بهینگی که بر مبنای کران پایین LB_{med} پایه گذاری شده، ارائه می شود.

گزاره ۵-۴([۳۳]): فرض کنید x_{med} یک نقطه بهینه مساله ۱-میانه وزن دار باشد و $\pi_{x_{med}}$ دنباله ای محاسبه شده با قانون ScheLoc ERD با استفاده از $r_i(X_{med})$ که $s\in\varphi$، باشد. آنگاه دنباله $\pi^* = \pi_{x_{med}}$ و مکان ماشین $x_{\pi^*} = x_{med}$ ، یک جواب بهینه برای مساله 1-P-ScheLoc TCP با زمان های پردازش برابر است، اگر حداقل یکی از شرایط زیر برقرار باشد.

$$a)\ C_{\pi(s)}(x_{med}) \le r_{\pi(s+1)}(x_{med}) \qquad \forall s\in\{1,\dots,n-1\}$$
$$b)\ LB_{med} = \sum_{j=1}^{n} C_{\pi(j)}(x_{med})$$

در گزاره (۵-۴) اثبات می شود[۳۳] که اگر شرط (a) برقرار باشد، آنگاه شرط (b) نیز برقرار است. یعنی $a \Rightarrow b$. حال اگر هیچ یک از شرایط بهینگی پیشین برقرار نباشد باید از الگوریتم های ابتکاری برای حل مساله ScheLoc استفاده کنیم.

باید توجه داشته باشیم که مساله 1-P-ScheLoc TCP با زمان های پردازش برابر، یک مساله بهینه سازی کلی است. چون تابع هدف $C_{sum}(x) := \min_{\pi\in\Pi_n}\left\{\sum_{j=1}^{n} C_{\pi(j)}(x)\right\}$ در حالت کلی روی R^2 نامحدب است. (مثال ۵-۲ را ببینید).

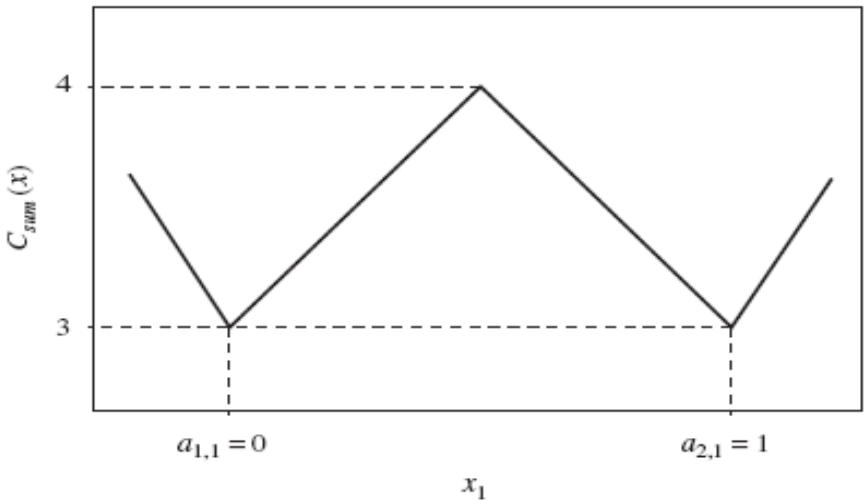
مثال ۵-۲: دو کار را با مکان های ذخیره سازی $a_1=(0,0)$ و $a_2=(15,0)$ و با نرم فاصله $(1\le q\le\infty)$ $d_i=\ell_q$ ، برای i=1,2 در نظر بگیرید. فرض کنید $p_1=p_2=p=1$ ، $v_1=v_2=1$, $\sigma_1=\sigma_2=0$ و $\Pi_2=\{(1,2),(2,1)\}$ باشد:

$$C_{sum}(a_1) = \min_{\pi\in\Pi_2}\left\{\sum_{j=1}^{n} C_{\pi(j)}(a_1)\right\} = (r_1(a_1)+p)+(\max\{r_2(a_1), r_1(a_1)+p\}+p)$$
$$= (1)+(\max\{1,1\}+1) = 3$$

---

[88] Weber

$$C_{sum}(a_2)=\min_{\pi\in\Pi_2}\left\{\sum_{j=1}^{n}C_{\pi(j)}(a_2)\right\}=(r_2(a_2)+p)+(\max\{r_1(a_2),r_2(a_2)+p\}+p)$$
$$=(1)+(\max\{1,1\}+1)=3$$
$$C_{sum}(0.5\times(a_1+a_2))=\min_{\pi\in\Pi_2}\left\{\sum_{j=1}^{n}C_{\pi(j)}(0.5\times(a_1+a_2))\right\}=(0.5+1)+(\max\{0.5,0.5+1\}+1)=4$$
$$>0.5\times(C_{sum}(a_1)+C_{sum}(a_2))=0.5\times(3+3)=3$$



شکل ۵-۲ شرح $C_{sum}(X)$ مثال ۵-۲

از این رو $C_{sum}(x)$ در حالت کلی روی R^2 نامحدب است (شکل ۵-۲([۳۳]) را ببینید).

بر مبنای این نتایج، در بخش بعدی الگوریتمی مفید برای حل مساله کل زمان اتمام کـار 1-P-ScheLoc و همین طور مساله 1-P-ScheLoc TCP با زمان های پردازش برابر، ارائه می شود.

۵-۲-۳ الگوریتم حل مساله

در این بخش هدف، مینیمم کردن توابع هدف ($C_{\max}(x)$ و $C_{sum}(x)$) با تقریـب مثلـث بـزرگ مثلث کوچک [۸۹] (BTST) است که توسط درزنر و سـوزوکی[۹۰] [۴۱] بـرای حـل مسـائل مکانیـابی نامحدب بر روی صفحه، پیشنهاد شده است. مکان های ذخیره سازی $a_1,\ldots,a_n$ داده شده است.

[89] The big triangle small triangle approach
[90] Suzuki

۵-۲-۳-۱ الگوریتم

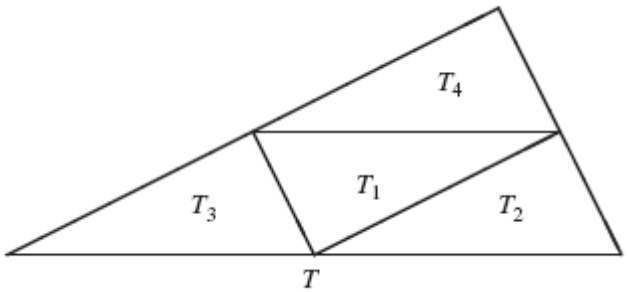
فرض کنید منطقه شدنی که شامل تعدادی متناهی چندضلعی محدب و یک نسبت درستی ε است، داده شده است.

گام۱- هر چند ضلعی محدب با استفاده از مثلث بندی دلانی[۹۱] ، مثلث بندی می شود [۴۲]. رئوس مثلث ها، مکان های ذخیره سازی کارها و رئوس چندضلعی های محدب هستند. اجتماع این مثلث ها، مجموعه آغازین مثلث های ζ را تشکیل می دهند.

گام۲- به ازای هر مثلث $T \in \zeta$، یک کران بالا برای مینیمم مقدار ممکن تابع هدف در T ، UB(T)، و یک کران پایین، LB(T) محاسبه کنید. کوچکترین UB(T) را $\overline{UB}$ بنامید. همه مثلث های $T' \in \zeta$ که $LB(T') \geq \overline{UB}(1-\varepsilon)$ را کنار بگذارید.

گام ۳- مثلث T با کوچکترین UB(T) را انتخاب کنید و آن را به چهار مثلث کوچک که یکی در وسط و بقیه در اطراف آن قرار گیرد، (شکل ۵-۳ را ببینید) تقسیم کنید. برای هر مثلث جدید $T_s \in \{T_1, T_2, T_3, T_4\}$، $UB(T_s)$ و $LB(T_s)$ را محاسبه کرده و $\overline{UB}$ را به روز کنید. حال قرار دهید $\zeta := \zeta \cup \{T_1, T_2, T_3, T_4\} \setminus \{T\}$. هر مثلث کوچک $T_s \in \{T_1, T_2, T_3, T_4\}$ که $LB(T_s) \geq \overline{UB}(1-\varepsilon)$ را از ζ کنار بگذارید. اگر $\overline{UB}$ تغییر کرده بود، تمام مثلث های $T' \in \zeta$ که $LB(T') \geq \overline{UB}(1-\varepsilon)$ را کنار بگذارید.

معیار توقف: عمل شاخه و کران زمانی متوقف می شود که هیچ مثلثی برای از دست دادن وجود نداشته باشد. یعنی، $\zeta = \phi$. آخرین نقطه ای که در آن $\overline{UB}$ بدست آمده است، جواب مساله می باشد. و مقدار $\overline{UB}$ با تقریب ε جواب بهینه می باشد.



شکل ۵-۳ تقسیم مثلث T به چهار مثلث جدید

[91] Delaunay

توجه کنید که:

۱- یک کران بالا برای مقدار مینیمم ممکن در مثلث، مقدار تابع هدف در هر نقطه مثلث می باشد (مانند مرکز ثقل).
۲- چون مثلث بندی بر مبنای مکان های ذخیره سازی به عنوان رئوس مثلث، پایه گذاری شده است، بنابراین هیچ مکان ذخیره سازی در داخل مثلث وجود ندارد. این مطلب برای تمام مثلث های تولید شده در ادامه فرایند، نیز صادق می باشد.
۳- به این ترتیب برای محاسبه کران پایین، نیازمند محاسبه کوتاهترین فاصله یک نقطه، که درون مثلث نیست، تا مثلث هستیم.
در زیر بخش بعدی چگونگی محاسبه کوتاهترین فاصله برای فاصله کلی ℓ_q، برای $q \geq 1$ نشان داده شده است.

۵-۲-۳-۲ کوتاهترین فاصله برای یک مثلث

در این زیر بخش کوتاهترین فاصله بین یک نقطه خارج مثلث و همه نقاط درون مثلث را پیدا می کنیم. گزاره زیر بدیهی است.

گزاره ۵-۵: کوتاهترین فاصله از یک مثلث، فاصله تا یک نقطه روی مرزش می باشد.

ابتدا کوتاهترین فاصله بین یک نقطه و یک خط را پیدا می کنیم. یک نقطه در صفحه با (u,v) نشان داده می شود. نقطه (u_1,v_1) و خط $v = mu + b$ را در نظر بگیرید. برای کامل کردن بحث، چگونگی یافتن نقطه ای روی خط که نزدیکترین نقطه به (u_1,v_1) باشد را نشان می دهیم. محاسبه چنین نقطه ای برای تحلیل مان الزامی است، زیرا رسیدن به اینکه نزدیکترین نقطه روی ضلع مثلث است یا نه، مورد نیاز است. مقدار u روی خط که نزدیکترین به (u_1,v_1) است را پیدا می کنیم. مقدار v می تواند با جایگذاری شدن در فرمول خط $v = mu + b$ پیدا شود. فاصله ℓ_q بین نقاط (u,v) و (u_1,v_1) به صورت زیر تعریف می شود:

$$d^q = |u-u_1|^q + |v-v_1|^q = |u-u_1|^q + |mu+b-v_1|^q$$
$$= |u-u_1|^q + |m|^q |u-b_1|^q \qquad (۱۲-۵)$$

که

$$b_1 = (v_1 - b)/m$$

۵-۲-۳-۲-۱ حالت $q>1$

در حالت $q>1$ گزاره زیر را داریم:

گزاره ۵-۶ ([۳۳]): مولفه بهینه u^*، که معادله (۵-۱۲) را مینیمم می کند از رابطه زیر بدست می آید:

$$\min\{u_1,b_1\}\le u^*\le \max\{u_1,b_1\}$$

اثبات: معادله (۵-۱۲) مجموع دو تابع محدب می باشد. مینیمم اولین تابع در u_1 و مینیمم دومین تابع در b_1 بدست می آید. بنابراین مینیمم مجموع این دو تابع، بین این دو مقدار می باشد زیرا دو تابع، هنگامی که u دور از انتهای پاره خط واصل u_1 و b_1 حرکت داده می شود، صعود می کنند.

فرض کنید $b_1>u_1$ باشد. حالت دیگر نیز همین نتیجه را می دهد. با گزاره (۵-۶)، $u_1<u^*<b_1$ می باشد. معادله ای که از مساوی با صفر قرار دادن فاصله بدست می آید، به صورت زیر می باشد:

$$\left(u^*-u_1\right)^{q-1}-|m|^q\left(b_1-u^*\right)^{q-1}=0 \quad\Rightarrow\quad u^*-u_1=|m|^{\frac{q}{q-1}}\left(b_1-u^*\right)$$

$$\Rightarrow\quad u^*=\frac{u_1+|m|^{\frac{q}{q-1}}b_1}{1+|m|^{\frac{q}{q-1}}} \qquad (۱۳-۵)$$

گزاره ۳-۷ ([۳۳]): جواب معادله (۵-۱۳) در رابطه $\min\{u_1,b_1\}\le u^*\le \max\{u_1,b_1\}$ صدق می کند.

اثبات: زیرا u^* ترکیب محدبی از u_1 و b_1 می باشد.

۵-۲-۳-۲-۲ حالت $q=1$

در حالت $q=1$ معادله (۵-۱۲)، مساله وبر روی خط با دو نقطه u_1 و b_1 می باشد. وقتی که $|m|\le 1$ باشد، جواب معادله $u^*=u_1$ می باشد. وقتی که $|m|\ge 1$، جواب معادله $u^*=b_1$ می باشد.

و هنگامی که $|m|=1$، پاره خط واصل دو نقطه جواب بهینه می باشد. توجه کنید که همین نتیجه برای رابطه (۵-۱۳) به صورت حد $q\to 1$ بدست می آید.

۵-۲-۳-۳ چگونگی روش

Sd ، کوتاهترین فاصله از مثلث، به صورت زیر محاسبه می شود.

۱- Sd را کوتاهترین فاصله تا سه راس قرار دهید.

۲- مراحل زیر را برای سه ضلع مثلث تکرار کنید:

a) u یی که فاصله را برای ضلع به کار رفته در رابطه (۵-۱۳) یا حالت $q=1$ مینیمم می کند، بیابید.

b) اگر مقدار u ، بین مولفه x دو راس آن ضلع باشد،

- فاصله d را تا نقطه روی ضلع محاسبه کنید.
- اگر $d<sd$، قرار دهید $sd=d$.

c) در غیر این صورت، از آن ضلع چشم پوشی می کنیم.

۵-۲-۴ معیارهای مکانیابی و زمانبندی برای الگوریتم جواب

به این ترتیب برای حل مسائل کل زمان اتمام کار و مجموع زمان های کامل شدن 1-P-ScheLoc تحت نرم های ℓ_q ، ($1\le q\le\infty$) با استفاده از تقریب مثلث بزرگ مثلث کوچک ، باید چند ضلعی یی را بیابیم که شامل جواب بهینه باشد و برای هر تابع کران های بالا و پایین را در مثلث مشخص کند.

۵-۲-۴-۱ معیار مجموعه مکانیابی غالب

در این زیر بخش نشان داده می شود که جواب بهینه برای مسائل کل زمان اتمام کار و مجموع زمان های کامل شدن 1-P-ScheLoc ، در داخل پوسته محدب مکان های ذخیره سازی قرار دارد. بنابراین می توان الگوریتم مثلث بزرگ مثلث کوچک را برای پوسته محدب مکان های ذخیره سازی به کار گرفت.

گزاره ۵-۸ ([۳۳]): جواب بهینه برای مسائل کل زمان اتمام کار و مجموع زمان های کامل شدن 1-P-ScheLoc ، در داخل پوسته محدب مکان های ذخیره سازی قرار دارد.

وندل[۹۲] و هارتر[۹۳] [۴۳] ثابت کردند که جواب بهینه مساله وبر با استفاده از فاصله ℓ_q ، ($1 \le q \le \infty$) ، در پوسته محدب مکان های ذخیره سازی می باشد. درزنر و گلدمن[۹۴] [۴۴] نشان دادند که برای نرم های ℓ_1 مکان های بهینه مساله وبر، یک زیر مجموعه از پوسته محدب مکان های موجود می باشد. اثبات وندل و هارتر [۴۳] بر پایه نقطه ای خارج پوسته محدب پایه گذاری شده است، نقطه ای در پوسته محدب وجود دارد که برای آن نقطه، فاصله تا همه مکان های موجود کمترین است . بنابراین چنین اثباتی می تواند برای بهینه سازی هر تابعی از فاصله ها برای مکان های موجود، که نسبت به فاصله ها به طور یکنواخت صعودی باشد، تعمیم یابد. که تابع هدف هر دو مدل یک تابع بطور یکنواخت صعودی از فاصله ها می باشد.

۵-۲-۴-۲ معیار کل زمان اتمام کار

۵-۲-۴-۲-۱ کران بالا $UB(T)$ برای مثلث $T \in \zeta$

فرض کنید $c(T)$ مرکز ثقل T باشد. آنگاه یک کران بالا برای مینیمم کل زمان اتمام کار در T، با تابع هدف محاسبه شده در $c(T)$ ، به صورت زیر می باشد:

$$, C_{\max}(T) := \min_{x \in T} \{C_{\max}(x)\} <= UB(T) := C_{\max}(c(T)) = C_{\alpha(n)}(c(T))$$

که دنباله α با استفاده از قانون ScheLoc ERD و برای مکان $c(T)$ تولید شده است یعنی،

$$r_{\alpha(1)}(c(T)) \le \ldots \le r_{\alpha(n)}(c(T))$$

۵-۲-۴-۲-۲ کران پایین $LB(T)$ برای مثلث $T \in \zeta$

---

[92] Wendell
[93] Hurter
[94] Goldman

کوتاهترین فاصله ممکن بین مکان ذخیره سازی a_i و هر نقطه در یـک مثلـث، $\ell_q^{\min}(a_i,T)$، را می توا ن به راحتی محاسبه کرد (بخش ۵-۳-۲ را ببینیـد). چـون بـه ازای هـر $i\in\varphi$، $\sigma_i\geq 0$ و $\tau_i>0$ ، کران های پایین را برای زمان های آماده سازی به صورت زیر محاسبه می کنیم:

$$r_i^{\min}(T):=\min_{x\in T}\{r_i(x)\}=\sigma_i+\tau_i\ell_q^{\min}(a_i,T)\qquad \forall i\in\varphi$$

حال دنباله α، با استفاده از قانون ScheLoc ERD برای $r_i^{\min}(T)$، $i\in\varphi$، به صورت زیـر بدسـت می آید:

$$r_{\alpha(1)}^{\min}(T)\leq\ldots\leq r_{\alpha(n)}^{\min}(T)$$

و در نهایت کران پایین برای مینیمم کل زمان اتمام کار به صورت زیر به دست می آید:

$$LB(T):=C_{\alpha(n)}^{\min}(T)=\max\left\{r_{\alpha(n)}^{\min}(T)+p_{\alpha(n)},r_{\alpha(n-1)}(T)+\sum_{s=n-1}^{n}p_{\alpha(s)},r_{\alpha(1)}^{\min}+\sum_{s=1}^{n}p_{\alpha(s)}\right\}$$

قضیه۵-۱ ([۳۳]): به ازای هر $T\in\zeta$، $LB(T)\leq C_{\max}(T)$.

اثبات: داریم:

$$\forall\pi\in\Pi_n\qquad\forall x\in T:\ r_{\pi(j)}^{\min}(T)\leq r_{\pi(j)}(x)\qquad\forall j\in\{1,\ldots,n\}$$

پس

$$C_{\pi(n)}^{\min}\leq C_{\pi(n)}$$

بنابراین

$$LB(T)=\min_{\mu\in\Pi_n}\{C_{\pi(n)}^{\min}(T)\}\leq C_{\pi(n)}^{\min}(T)\leq C_{\pi(n)}(x)$$

و در نتیجه

$$LB(T)\leq\min_{\pi\in\Pi_n,x\in T}C_{\pi(n)}(x)=C_{\max}(T)$$

توجه شود که اگر در تقریب مثلث بزرگ مثلث کوچک، $\phi=\zeta$، آنگـاه یـک مکـان $x(T^*)$ و یـک دنباله π^* برای مساله کل زمان اتمام کار 1-P-ScheLoc، پیدا کرده ایم، که در مورد آن می دانـیم که $\overline{UB}=UB(T^*)$، با تقریب ε می باشد.

۵-۲-۴-۳ معیار مجموع زمان های کامل شدن

۵-۲-۴-۳-۱ کران بالا $UB(T)$ برای مثلث $T \in \zeta$

فرض کنید $c(T)$ مرکز ثقل T باشد. آنگاه یک کران بالا برای مینیمم مجموع زمان هـای کامل شدن به صورت زیر می باشد:

$$C_{sum}(T) := \min_{x \in T} \{C_{sum}(x)\} <= UB(T) := C_{sum}(c(T)) = \sum_{j=1}^{n} C_{\alpha(j)}(c(T))$$

که دنباله α با استفاده از قانون ScheLoc ERD و برای مکان $c(T)$ تولید شده است.یعنی

$$r_{\alpha(1)}(c(T)) \le \ldots \le r_{\alpha(n)}(c(T))$$

۵-۲-۴-۳-۲ کران پایین $LB(T)$ برای مثلث $T \in \zeta$

کوتاهترین فاصله ممکن بین مکان ذخیره سازی a_i و هر نقطه در یک مثلـث، $\ell_q^{\min}(a_i, T)$، را می توا ن به راحتی محاسبه کرد (بخش ۵-۳-۲ را ببینید). چون بـه ازای هـر $i \in \varphi$، $\sigma_i \ge 0$ و $\tau_i > 0$ می باشد، کران های پایین را برای زمان های آماده سازی به صورت زیر محاسبه می کنیم:

$$r_i^{\min}(T) := \min_{x \in T} \{r_i(x)\} = \sigma_i + \tau_i \ell_q^{\min}(a_i, T) \qquad \forall i \in \varphi$$

حال دنباله α با استفاده از قانون ERD برای $r_i^{\min}(T)$، $i \in \varphi$، به صورت زیر بدست می آید:

$$r_{\alpha(1)}^{\min}(T) \le \ldots \le r_{\alpha(n)}^{\min}(T)$$

و در نهایت با استفاده از روابط

$$C_{\alpha(1)}^{\min}(T) = r_{\alpha(1)}^{\min}(T) + p$$
$$C_{\alpha(j)}^{\min}(T) = \max\left\{C_{\alpha(j-1)}^{\min}(T), r_{\alpha(j)}^{\min}(T)\right\} + p \qquad \forall j \in \{2,\ldots,n\}$$

کران پایین برای $C_{sum}(T)$ به صورت زیر است:

$$LB(T) := \sum_{j=1}^{n} C_{\alpha(j)}^{\min}(T)$$

قضیه ۵-۲ ([۳۳]): به ازای هر $T \in \zeta$ ، $LB(T) \le C_{sum}(T)$.

اثبات: داریم

$$\forall \pi \in \Pi_n \qquad \forall x \in T:\ r_{\pi(j)}^{\min}(T) \le r_{\pi(j)}(x) \qquad \forall j \in \{1,...,n\}$$

پس

$$C_{\pi(j)}^{\min} \le C_{\pi(j)} \qquad \forall j \in \{1,...,n\}$$

بنابراین

$$LB(T) = \min_{\mu \in \Pi_n} \left\{ \sum_{j=1}^{n} C_{\mu(n)}^{\min}(T) \right\} \le \sum_{j=1}^{n} C_{\pi(j)}^{\min}(T) \le \sum_{j=1}^{n} C_{\pi(n)}(x)$$

و در نتیجه

$$LB(T) \le \min_{\pi \in \Pi_n, x \in T} \sum_{j=1}^{n} C_{o(n)}(x) = C_{sum}(T)$$

توجه کنید که اگر در تقریب مثلث بزرگ مثلث کوچک، $\zeta = \phi$ باشد، آنگاه یک مکان $x(T^*)$ و یک دنباله π^* برای مساله 1-P-ScheLoc TCP با زمان های پردازش برابر پیدا کرده ایم، که در مورد آن می دانیم که $\overline{UB} = UB(T^*)$، با تقریب ε بهینه می باشد.

۵-۲-۵ نتایج عددی

برنامه ها در فورترن[۹۵] وابسته به علم حساب و با دقت زیاد تدوین شده است. و روی کامپیوتر پنتیوم ۵ با حافظه ۲۵۰ مگا بایت اجرا شده است. ده مساله با $n = 10,20,50,...10000$ ایجاد شده است. با مکان های ذخیره سازی که در $[-500,500]\times[-500,500]$ قرار گرفته اند و زمان های فرارسی ذخیره $\sigma_i \in [0,2500]$ و سرعت انتقال $v_i \in [0.5,1]$ به ازای هر $i \in \varphi$. برای مساله کل زمان اتمام کار ScheLoc ، $p_i \in [1,50]$ و برای مساله مجموع زمان های کامل شدن ScheLoc ، $p_i = p = 25$ ، به ازای هر $i \in \varphi$ انتخاب شده است. و نسبت درستی $\varepsilon = 10^{-5}$ در نظر گرفته شده است.

مقدار مینیمم، ماکزیمم و میانگین برای هر یک از اندازه های: تعداد تکرارها، ماکزیمم مثلث ها در طی فرایند شاخه و کران و مجموع زمان در هر اجرا به ثانیه در جدول (۵-۱) و (۵-۲) [۳۳]

---

[95] Fortran

گزارش شده است. توجه کنید که تعداد مثلث ها در پایان مثلث بندی برابر $2n-k-2$ مـی باشـد که k تعداد راس های پوسته محدب نقاط $\{a_1,a_2,...,a_n\}$ می باشد [۴۱]. در این تجربه محاسباتی تعداد مثلث ها خیلی نزدیک به $2n$ بود (یعنی برای بزرگترین مساله کمتر از 20,000). از ایـن رو فقط ماکزیمم تعداد مثلث ها در طـی فراینـد شـاخه و کـران گـزارش شـده اسـت. هـر دو مسـاله ScheLoc با فاصله های خط راست (اقلیدسی) (q=2)، مسـتطیلی (q=1) و ℓ_q بـا (q=1.78) حـل شده است.

جدول۵-۱ نتایج محاسبات برای کل زمان اتمام کار

n	تکرارها			ماکزیمم مثلث ها			زمان(ثانیه)		
	min.	ave.	max.	min.	ave.	max.	min.	ave.	max.
Euclidean distances ($q=2$)									
10	63	2142	614.2	19	570	147.3	0.00	0.02	0.01
20	51	159	97.7	21	75	36.9	0.00	0.00	0.00
50	59	511	128.1	19	113	41.4	0.00	0.03	0.01
100	52	1310	252.4	25	296	76.5	0.02	0.22	0.05
200	45	480	149.3	21	110	55.5	0.06	0.22	0.10
500	46	140	84.1	20	86	49.2	0.33	0.42	0.36
1000	37	125	71.3	30	69	47.1	1.39	1.61	1.52
2000	14	91	48.0	21	74	40.5	6.41	7.45	6.96
5000	8	49	23.0	13	35	24.2	55.94	65.08	61.20
10,000	7	22	11.9	9	26	13.3	266.84	313.89	290.37
Rectilinear distances ($q=1$)									
10	10,291	190,305	62,760.8	2894	50,908	18,160.8	0.16	34.98	7.37
20	1373	134,063	35,598.8	339	42,465	11,044.4	0.03	22.41	3.72
50	420	70,212	19,635.7	110	22,464	5817.1	0.03	10.02	2.26
100	278	39,630	9687.6	77	11,619	2815.7	0.06	8.72	2.04
200	78	22,416	4103.4	29	7330	1305.2	0.08	9.66	1.84
500	91	3043	699.5	29	860	222.0	0.39	3.75	1.11
1000	40	184	95.0	23	90	63.3	1.47	1.84	1.67
2000	35	144	76.6	31	101	53.1	6.81	8.16	7.44
5000	15	120	42.1	15	79	35.7	58.27	68.78	63.23
10,000	5	36	16.1	12	32	21.8	274.19	323.66	298.41
ℓ_q distances ($q=1.78$)									
10	77	2576	650.1	25	685	163.0	0.00	0.14	0.04
20	54	288	131.1	25	86	42.9	0.00	0.03	0.02
50	61	1828	275.9	20	451	78.0	0.02	0.52	0.09
100	61	1472	291.8	24	365	84.8	0.08	0.91	0.21
200	65	584	175.8	27	119	57.3	0.20	0.88	0.35
500	42	142	94.9	26	124	57.6	0.95	1.30	1.13
1000	36	193	67.7	27	93	48.6	3.66	4.81	3.98
2000	19	73	44.9	24	66	43.7	15.42	16.67	16.02
5000	16	71	30.4	15	36	24.2	110.88	121.12	116.63
10,000	5	17	10.1	10	21	13.9	485.42	534.11	510.27

جدول۵-۲ نتایج محاسبات برای مجمع زمان های اتمام کار

n	تکرارها			ماکزیمم مثلث ها			زمان(ثانیه)		
	min.	ave.	max.	min.	ave.	max.	min.	ave.	max.
Euclidean distances ($q = 2$)									
10	2920	75,922	27,772.1	1165	38,603	13,236.8	0.03	8.69	2.33
20	699	41,099	7829.4	242	20,444	3557.0	0.02	2.92	0.40
50	365	5553	2498.6	108	2780	1148.1	0.02	0.45	0.19
100	218	1566	773.6	105	592	306.4	0.05	0.34	0.16
200	152	1469	467.4	71	655	191.3	0.11	0.72	0.24
500	84	543	231.1	51	203	112.4	0.38	0.88	0.54
1000	99	464	242.2	74	223	132.8	1.67	2.84	2.06
2000	35	390	137.8	42	208	91.4	6.67	9.16	7.72
5000	38	216	94.0	33	131	64.9	54.48	70.28	62.15
10,000	21	115	45.3	23	98	44.5	272.47	341.62	300.92
Rectilinear distances ($q = 1$)									
10	1951	75,399	13,005.6	596	36,542	5684.7	0.02	8.83	0.99
20	777	31,943	6731.2	319	12,223	2489.0	0.02	1.86	0.27
50	584	4743	2051.6	236	1875	718.0	0.05	0.38	0.16
100	372	1866	903.8	174	588	322.8	0.08	0.39	0.19
200	211	1017	563.1	111	351	213.2	0.14	0.48	0.29
500	124	784	323.8	81	325	160.0	0.45	1.19	0.68
1000	141	412	256.6	91	203	133.1	1.78	2.69	2.17
2000	63	503	242.3	30	285	121.1	7.16	10.55	8.60
5000	36	290	122.4	33	162	88.1	56.22	70.22	63.30
10,000	24	92	56.5	25	154	62.7	279.06	337.31	302.87
ℓ_q distances ($q = 1.78$)									
10	997	247,310	46,818.9	255	157,309	26,273.0	0.06	130.92	16.29
20	779	26,556	6506.3	242	12,401	2724.4	0.08	3.78	0.83
50	257	4181	1501.1	83	1731	523.1	0.08	1.23	0.44
100	263	1047	561.1	105	323	189.1	0.19	0.70	0.38
200	199	678	375.4	68	260	151.7	0.38	0.98	0.61
500	94	471	262.8	56	200	105.3	1.16	2.38	1.69
1000	72	469	244.7	52	205	116.8	3.97	6.84	5.26
2000	29	299	151.2	39	129	86.6	15.52	19.92	17.65
5000	30	230	98.3	33	122	71.2	109.69	124.14	117.45
10,000	16	127	48.4	17	98	41.3	485.97	540.61	511.56

۳-۵ مساله توام زمانبندی و مکانیابی تک ماشینی بر روی خط

دراین بخش ما حالتی از مساله زمانبندی-مکانیابی را در نظر می گیریم که ماشین فقط روی یک خط مشخص(L-ScheLoc) قرار گیرد.

یکی از کاربردهای این مساله لنگر انداختن کشتی در لنگرگاهی است که به صورت خط می باشد و بار گیری کانتینرها که هر یک در مکانی از بندر قرار گرفته اند مورد نظر می باشد، به گونه ای که مجموع زمان اتمام بارگیری یا کل زمان اتمام بارگیری کشتی حداقل شود. در این مساله می خواهیم بهترین مکان برای کشتی را روی خط لنگر گاه پیدا کنیم (مکانیابی) و چگونگی فرایند بارگیری کانتینرها (زمانبندی) را مشخص کنیم. بنابراین کانتینرها روی صفحه قرار دارند ولی کشتی فقط می تواند روی یک خط حرکت کند .

تعریف مساله: فرض کنید $\varphi=\{1,...,n\}$ یک مجموعه از کارها باشد که روی صفحه قرار گرفته اند. هر کار دارای زمان پردازش نامنفی p_i که $i\in\varphi$ است که باید به صورت غیر قابل قطع روی تک ماشین M زمانبندی شود به گونه ای که M فقط می تواند روی خط $L:bx+dy=c$ حرکت کند. به هر کار $i\in\varphi$ مکان ذخیره سازی a_i در R^2 نسبت داده می شود. از این رو مساله توام زمانبندی مکانیابی تک ماشینی روی خط (1-L-ScheLoc) انتخاب مکان ماشین M یعنی نقطه $a\in L$ تحت شرایطی است که مجموعه کارهای φ کاملاً پردازش شوند. هدف ما بهینه کردن تابع هدف زمانبندی کل زمان اتمام کار و مجموع زمان های کامل شدن است که نه تنها به دنباله انجام کارها وابسته است بلکه به انتخاب a نیز بستگی دارد. همانند بخش ۵-۲ زمان فرارسی ذخیره سازی $\sigma_i\geq 0$ معادل است با زمانی که کارi در مکان ذخیره سازی اش قابل دسترس می باشد.

چون σ_i مثبت است بنابراین می تواند در طی پردازش کارهای دیگر حرکت داده شود. سرعت انتقال $v_i>0$، سرعت حرکت کار ***i*** تا M می باشد. فرض کنید تابع فاصله $d_i(a_i,a)$ یک تابع فاصله کلی روی R^2 باشد آنگاه با فرض $\tau_i=1/v_i$، $r_i(a)=\sigma_i+\tau_i d_i(a_i,a)$ زمان جابجایی یا متغیر زمان آماده سازی کار ***i*** تا ماشین M می باشد. دنباله کارهایی که روی ماشین پردازش می شوند با یک جایگشت π از $\{1,...,n\}$ تعیین می شوند که $\pi(j)=i$ به این معنی است که کار ***j*** ،***i*** امین کاری است که پردازش می شود. مجموعه تمام جایکشت های $\{1,...,n\}$ را با Π_n نشان می

دهیم. به ازای هر دنباله $\pi \in \Pi_n$ و هر مکان $a \in L$ برای ماشین، زمان کامل شدن به ازای هر کار $i \in \varphi$ با استفاده از رابطه بازگشتی زیر تعریف می شود:

$$C_{\pi(1)}(a) = r_{\pi(1)}(a) + p_{\pi(1)} \qquad (۱۴-۵)$$

$$C_{\pi(j)}(a) = \max\left\{C_{\pi(j-1)}(a), r_{\pi(j)}(a)\right\} \qquad j \in \{2,3,\ldots,n\} \qquad (۱۵-۵)$$

و $p_{\pi(j)}$ زمان پردازش کار $\pi(j)$ تعریف می شود. فرمولهای (۵-۱۴) و (۵-۱۵) ، به ازای هر $j \in \{1,2,\ldots,n\}$. می توانند به صورت زیر بیان شوند:

$$c_{\pi(j)}(a) = \max\{r_{\pi(j)}(a) + p_{\pi(j)}, r_{\pi(j-1)} + \sum_{s=j-1}^{j} p_{\pi(s)}, \ldots, r_{\pi(1)} + \sum_{s=1}^{j} p_{\pi(s)}\} \qquad (۱۶-۵)$$

و در نهایت ماکزیمم زمان کل اتمام کار در $a \in L$ و برای یک $\pi \in \Pi_n$ به صورت زیر نوشته می شود:

$$\max\{c_1(a), \ldots, c_n(a)\} = c_{\pi(n)}(a) \qquad (۱۷-۵)$$

و مجموع زمان های کامل شدن در $a \in L$ و برای یک $\pi \in \Pi_n$ به صورت زیر می باشد:

$$\sum_{s=1}^{n} c_s(a) = \sum_{j=1}^{n} c_{\pi(j)}(a) \qquad (۱۸-۵)$$

حال به بررسی این دو مساله خاص در بخش های بعد می پردازیم.

۵-۳-۱ تحلیل کل زمان اتمام کار مساله 1-L-ScheLoc

در این بخش ما کل زمان اتمام کار مساله 1-L-ScheLoc را مورد بررسی و تحلیل قرار می دهیم. با استفاده از (۵-۱۶) و (۵-۱۷) تابع هدف را می توان به صورت زیر نوشت:

$$\min c_{\pi(n)}(a) = \max\{r_{\pi(n)}(a) + p_{\pi(n)}, r_{\pi(n-1)}(a) + \sum_{s=n-1}^{n} p_{\pi(s)}, \ldots, r_{\pi(1)}(a) + \sum_{s=1}^{n} p_{\pi(s)}\}$$

$$s.t \quad a \in L \qquad (۱۹-۵)$$

$$\pi \in \Pi_n$$

اگربرای تمام $i \in \varphi$، $p_i = 0, \sigma_i = 0, \tau_i = 0$ این مساله به یک مساله مکانیابی ۱-مرکزی ساده بر روی خط تبدیل می شود. مساله کل زمان اتمام کار 1-L-ScheLoc بعد از یافتن مکان ماشین، به مساله زمانبندی کل زمان اتمام کار کلاسیک با مدت زمان های آماده سازی ثابت یعنی

$r_i(a) = r$ که $i \in \varphi$ تبدیل می شود. در این حالت می توانیم از قانون زودترین زمان آماده سازی (ScheLoc ERD) استفاده می کنیم. یعنی دنباله کارها در ترتیب غیر نزولی مدت زمان آماده سازی برای بدست آوردن یک دنباله کاری بهینه قرار می گیرند یعنی:

$$r_{\pi(1)}(a) \le \ldots \le r_{\pi(n)}(a)$$

بنابراین متغیرهای $\pi \in \Pi_n$ در تابع هدف تنها به ترتیب غیر نزولی مدت زمان آماده سازی وابسته هستند و مدت زمان آماده سازی تنها به متغیر $a \in L$ وابسته است. با استفاده از این نتایج به فرمولبندی جدیدی به صورت زیر می رسیم:

$$\begin{aligned} &\min c_{\pi(n)}(a) = \max\{r_{\pi(n)}(a) + p_{\pi(n)}, r_{\pi(n-1)}(a) + \sum_{s=n-1}^{n} p_{\pi(s)}, \ldots, r_{\pi(1)}(a) + \sum_{s=1}^{n} p_{\pi(s)}\} \\ &s.t \quad r_{\pi(1)}(a) \le \ldots \le r_{\pi(n)}(a) \\ &\qquad a \in L \\ &\qquad \pi \in \Pi_n \end{aligned} \qquad (۲۰-۵)$$

یک کران پایین برای مساله به صورت $LB_{\max} = \min\{\beta_i\} + \sum p_i$ است، که $\beta_i = \sigma_i + \tau_i d_i(L, a_i)$.

لم۵-۱ فرض کنید $i \in \arg\min\{\beta_s \; s = 1,\ldots,n\}$ و $a = \alpha_i$ که $\alpha_i = d_i(L, a_i)$ (تصویر نقطه a_i روی خط L) و $\pi_{a_i} = (\pi(1) = i, \pi(2), \ldots, \pi(n))$ یک دنباله محاسبه شده با قانون ScheLoc ERD باشد آنگاه دنباله $\pi^* = \pi_{a_i}$ و مکان ماشین $a_{\pi^*} = \alpha_i$ یک جواب بهینه برای کل زمان اتمام کار مساله 1-L-ScheLocاست، اگر حداقل در یکی از شرایط زیر صدق کند:

$$\begin{aligned} &a)\ \beta_i + p_i \ge r_s(\alpha_i) = \sigma_i + \tau_s\, d_i(a_s, \alpha_i) && s \in \{1,\ldots,n\} \\ &b)\ \beta_i + \sum_{j=1,\ldots,s} p_{\pi(j)} \ge r_{\pi(s+1)}(\alpha_i) && s \in \{1,\ldots,n-1\} \\ &c)\ LB_{\max} = c_{\pi(n)}(\alpha_i) \end{aligned}$$

حال فرض می کنیم هیچ یک از شرایط لم ۵-۱ برقرار نیست بنابراین الگوریتم مفیدی برای حل مساله ارائه خواهیم داد. باید در نظر داشته باشیم که کل زمان اتمام کار مساله 1-L-ScheLoc یک مساله بهینه سازی کلی است و در حالت کلی روی R^2 نامحدب است.

۵-۳-۲ تحلیل مجموع زمان های اتمام کار مساله 1-L-ScheLoc

در این بخش ما مجموع زمان های اتمام کار مساله 1-L-ScheLoc را با زمان های پردازش برابر، یعنی به ازای $i \in \varphi$، $p_i = p$ باشد، مورد بررسی و تحلیل قرار می دهیم. با استفاده از (۵-۱۴)، (۵-۱۵) و (۵-۱۶) تابع هدف را می توان به صورت زیر نوشت:

$$\begin{aligned}
&\min \sum_{j=1}^{n} c_{\pi(j)}(a) \\
&s.t \quad c_{\pi(1)}(a) = r_{\pi(1)}(a) + p \\
&c_{\pi(j)}(a) = \max\{c_{\pi(j-1)}(a), r_{\pi(j)}(a)\} + p \qquad j \in \{2,...,n\} \qquad (۲۱-۵) \\
&a \in L \\
&\pi \in \Pi_n
\end{aligned}$$

اگربرای تمام $i \in \varphi$، $p = ۰, \sigma_i = ۰, \tau_i = ۰$ این مساله به یک مساله مکانیابی ۱-میانه ساده بر روی خط تبدیل می شود. مساله مجموع زمان اتمام کار 1-L-ScheLoc بعد از یافتن مکان ماشین، به مساله مجموع زمان اتمام کار کلاسیک با مدت زمان های آماده سازی ثابت یعنی $r_i(a) = r$ که $i \in \varphi$ تبدیل می شود. در این حالت می توانیم از قانون زودترین زمان آماده سازی (ScheLoc ERD) استفاده می کنیم. یعنی دنباله کارها در ترتیب غیر نزولی مدت زمان آماده سازی برای بدست آوردن یک دنباله کاری بهینه قرار می گیرند یعنی:

$$r_{\pi(1)}(a) \le ... \le r_{\pi(n)}(a)$$

بنابراین متغیرهای $\pi \in \Pi_n$ در تابع هدف تنها به ترتیب غیر نزولی مدت زمان آماده سازی وابسته اند و مدت زمان آماده سازی تنها به متغیر $a \in L$ وابسته است. با استفاده از این نتایج به فرمولبندی جدیدی به صورت زیر می رسیم:

$$\begin{aligned}
&\min \sum_{j=1}^{n} c_{\pi(j)}(a) \\
&s.t \quad c_{\pi(j)}(a) \ge c_{\pi(j-1)}(a) + p \qquad j \in \{2,...,n\} \\
&\qquad c_{\pi(j)}(a) \ge r_{\pi(j)} + p \qquad j \in \{1,...,n\} \qquad (۲۲-۵) \\
&\qquad a \in L \\
&\qquad \pi \in \Pi_n
\end{aligned}$$

یک کران پایین برای مساله به صورت $LB_{sum} := \sum_{j=1}^{n}(\min_{i\in p}\{\beta_i\} + \sum_{i=1}^{j} p)$ است، که $\beta_i = \sigma_i + d_i(L, a_i)$.

لم۵-۲: فرض کنید $i \in \arg\min\{\beta_s \; s = 1,\ldots,n\}$ و $a = \alpha_i$ که $\alpha_i = d_i(L, a_i)$(تصویر نقطه a_i روی خط L) و $\pi_{a_i} = (\pi(1) = i, \pi(2),\ldots,\pi(n))$ یک دنباله محاسبه شده با قانون ScheLoc ERD باشد آنگاه دنباله $\pi^* = \pi_{a_i}$ و مکان ماشین $a_{\pi^*} = \alpha_i$ یک جواب بهینه برای کل زمان اتمام کار مساله 1-L-ScheLocاست، اگر حداقل در یکی از شرایط زیر صدق کند:

$$a)\; c_{\pi(1)}(a_i) = \beta_i + p \geq r_s(\alpha_i) = \sigma_i + \tau_s \; dist(a_s, \alpha_i) \qquad s \in \{1,\ldots,n\}$$
$$b) c_{\pi(s)} \geq r_{\pi(s+1)}(\alpha_i) \qquad s \in \{1,\ldots,n-1\}$$
$$c)\; LB_{sum} = \sum_{j=1}^{n} c_{\pi(j)}(\alpha_i)$$

حال فرض می کنیم هیچ یک از شرایط لم ۵-۲ برقرار نیست بنابراین الگوریتم مفیدی برای حل مساله ارائه می دهیم. باید در نظر داشته باشیم که مجموع زمان های اتمام کار مساله 1-L-ScheLoc یک مساله بهینه سازی کلی است و تابع هدف $c_{sum}(a) := \min_{\pi\in\Pi_n}\{\sum_{j=1}^{n} c_{\pi(j)}(a)\}$ در حالت کلی روی R^2 نامحدب است.

در بخش بعد الگوریتمی برای حل مساله کل زمان اتمام کار 1-L-ScheLoc و همین طور مساله 1-L -ScheLoc TCP با زمان های پردازش برابر ارائه می شود. که این الگوریتم الگوریتمی ابتکاری است و برای حل مسائل نامحدب کاربرد دارد.

۵-۳-۳ الگوریتم

هدف ما کمینه کردن تابع $c_{sum}(a)$ و $c_{max}(a)$ با تقریب بازه بزرگ بازه کوچک می باشد. فرض کنید منطقه شدنی شامل تعدادی متناهی بازه روی خط $L: bx + dy = c$، a_i مکان ذخیره سازی کارها که $i \in \varphi$ و نسبت درستی $\varepsilon > 0$ باشد. فرض می کنیم خط L سمت راست محور x های صفحه مختصات R^2 باشد (در غیر این صورت با یک تبدیل می توان خط L را روی محور x ها منطبق کرد و به دنبال آن نقاط a_i را نیز انتقال می دهیم). حال اگر a_i را به صورت $a_i = (x_i, y_i)$ در نظر بگیریم آنگاه تصویر هر نقطه روی L برابر x_i می باشد. مجموعه آغازین بازه ها را γ در نظر می گیریم. حال گام های الگوریتم را به صورت زیر در نظر می گیریم:

گام ۱- تصویر نقاط روی خط L رامحاسبه کنید و تصویر هر نقطه را x_i در نظر بگیرید و سپس x_i ها را مرتب کنید.

گام۲- بازه ها را به صورت $I_i = [x_{i-1}, x_i]$ در نظر بگیرید .

گام۳- یک کران بالا برای مقدار مینیمم تابع هدف در I_i ، $UB(I_i)$ و یک کران پایین $LB(I_i)$ به ازای هر $I_i \in \gamma$ محاسبه کنید. فرض کنید کوچکترین $UB(I_i)$ ، $\overline{UB}$ باشد. همه بازه هایی را که $LB(I_i) \geq \overline{UB}(1-\varepsilon)$ را کنار می گذاریم.

گام ۴- بازه I_i را باکوچکترین $UB(I_i)$ انتخاب کنید و آن را به سه بازه مساوی تقسیم کنید و $UB(I_{i_s})$ و $LB(I_{i_s})$ را برای هر بازه جدید $I_{i_s} \in \{I_{i_1}, I_{i_2}, I_{i_3}\}$ محاسبه کنید و $\overline{UB}$ را به روز رسانی کنید. قرار دهید $\gamma := \gamma \cup \{I_{i_1}, I_{i_2}, I_{i_3}\} \setminus \{I_i\}$. هر بازه $I_{i_s} \in \{I_{i_1}, I_{i_2}, I_{i_3}\}$ که $LB(I_{i_s}) \geq \overline{UB}(1-\varepsilon)$ از γ کنار گذاشته می شود. اگر $\overline{UB}$ تغییر کرده باشد، تمام بازه های $I'_i \in \gamma$ ارزیابی می شوند و تمام بازه هایی که $LB(I') \geq \overline{UB}(1-\varepsilon)$ کنار گذاشته می شوند.

معیار توقف: شاخه و کران هنگامی متوقف می شود که هیچ بازه ای باقی نمانده باشد یعنی $\gamma = \phi$. آخرین نقطه ای که در آن $\overline{UB}$ بدست آمده است، جواب مساله می باشد. و مقدار $\overline{UB}$ با تقریب ε نسبت به جواب بهینه می باشد.

توجه کنید که: اولا فرض بر این است که هیچ یک از مکان های ذخیره سازی بر روی خط L نباشند. بنابراین هیچ یک از مکان های ذخیره سازی در داخل بازه ها نمی باشند. ثانیا یک کران بالا برای مقدار مینیمم ممکن در بازه، مقدار تابع هدف در هر نقطه بازه می باشد (مانند مرکز بازه).

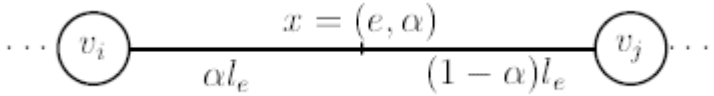
فصل ۶
مساله توام زمانبندی مکانیابی بر روی شبکه

۶-۱ مقدمه

در این فصل مساله توام مکانیابی تک وسیله ای شبکه ای و زمانبندی تک ماشینی را بررسی می کنیم [۲۸]. که هر دو زمینه در تحقیق در عملیات بسیار شناخته شده هستند و معمولا مجزا از یکدیگر مورد بحث قرار گرفته اند. شاید بدین خاطر است که مباحث مکانیابی اغلب مسائلی استراتژیک و مباحث زمانبندی مباحث عملیاتی است اما مثالهایی در طرحهای صنعتی و علم کامپیوتر نشان می دهد که این مطلب لزوما صحیح نمی باشد. برای نمونه، استفاده از ماشین های متحرک، مثالی است که مکانیابی مثل زمانبندی بحثی عملیاتی می باشد. طراحی مکان ماشین توام با سازماندهی خط تولید بهره وری را افزایش می دهد. به بیان دیگر مکانیابی و زمانبندی ترکیب می شود. گراف $G=(V,E)$ با مجموعه گره های $V=\{v_1,...v_n\}$ و مجموعه یالهای $E=\{e_1,...,e_n\}$ را در نظر بگیرید. گره ها نشان دهنده کارهای $A_1,...,A_n$ که هر یک دارای زمان پردازش P_j، $j=1,...,n$ می باشد، هستند. یال $e=(v_i,v_j)$ امکان حرکت از کار A_i به کار A_j مستقیما در $L_e=L_{ij}$ واحد زمان (یا بر عکس) را ارائه می دهد. اگر هیچ یالی بین A_i و A_j وجود نداشته باشد، از میان کارهایی که کوتاهترین مسیر از V_i به V_j نسبت به طول یال L_e را می دهد، حرکت می کنیم. کوتاهترین فاصله بین هر یال V_iو V_j در V در زمان $O(n^3)$ می تواند محاسبه شود [۴۵]. بخش مکانیابی مساله پاسخ به این سوال است که ماشینی که باید تمام کارهای A_1و ... و A_n بر روی آن پردازش گردد، کجا قرار بگیرد. در **گونه راسی**[۹۶] مساله مکانیابی- زمانبندی، مکانیابی فقط روی راس ها صورت می گیرد، در حالی که در **گونه مطلق**[۹۷] مکانیابی- زمانبندی، ماشین می تواند هر جای گراف قرار بگیرد یعنی، روی یالها نیز می تواند قرار گیرد. اگر $e=[v_i,v_j]\in E$ باشد، آنگاه $x=(e,\alpha)$ با $\alpha\in[0,1]$ نقطه ای در G است که:

$$d(v_i,x)=d(x,v_i)=\alpha L_e \quad \text{و} \quad d(v_j,x)=d(x,v_j)=(1-\alpha)L_e \qquad (۱-۶)$$

(شکل ۶-۱را ببینید) یعنی کیفیت خطی بودن فاصله در طول یال e در نظر گرفته می شود. مجموعه نقاط در G با χ نشان داده می شود.



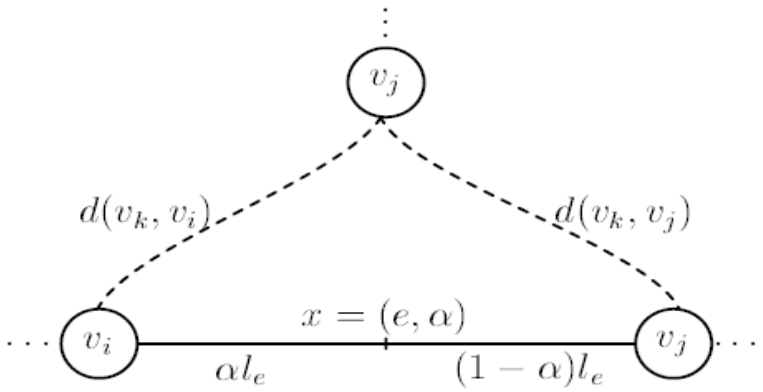
شکل ۶-۱: نمایش نقطه $x=(e,\alpha)$ بر روی یال $e=[v_i,v_j]$

[96] node version
[97] absolute version

هر بار که مکانی برای ماشین انتخاب می شود، مساله زمانبندی- مکانیابی به یک مساله زمانبندی کلاسیک با زمان های آماده سازی، کاهش می یابد. ماشین در آن واحد می تواند تنها یکی از کارهای $A_1,...,A_n$ را پردازش کند. در طی پردازش، حق تقدم مجاز نمی باشد. برای انتقال هر کار A_k به مکان ماشین x داریم:

$$d(v_k,x)=\begin{cases} d_{kl} & if\ x=v_l\in V \\ \min\{d(v_k,v_i)+\alpha l_{ij} & if\ x=(e,\alpha)=\chi\setminus V,\ e=[v_i,v_j] \\ \quad d(v_k,v_j)+(1-\alpha)l_{ij}\} & \end{cases} \qquad (۲-۶)$$

در نتیجه $r_k(x)=d(v_k,x)$ ، زمان آماده سازی کار A_k است و پردازش کار A_k نمی تواند قبل از این زمان آغاز گردد.



شکل ۶-۲: تجسم زمان آماده سازی $r_k(x)=d(v_k,x)$ برای کار A_k، اگر ماشین در $x\in\chi$ قرار داشته باشد

در این پایان نامه به بررسی کل زمان اتمام کار، یعنی زمان کامل شدن ماکسیمال کارها، یا $C_{\max}=\max\{C_i(x),\ i\in\{1,...,n\}\}$ می پردازیم که $C_i(x)$ زمانی است که پردازش کار A_i ام کامل می شود.

هر زمانبندی با یک دنباله ای مرتب از کارها که با $\Pi=(\pi(1),...,\pi(n))$ نشان داده می شود، مشخص می شود داریم $C_{\max}(x)=C_{\pi(n)}(x)$ که شکل کلاسیک مساله به صورت $1|\ r_j\ |\ C_{\max}$ می باشد و این مساله با مرتب سازی کارها در ترتیب غیر نزولی از زمان آماده سازی شان (قانون زودترین زمان آماده سازی (ERD)) می تواند حل گردد. برخلاف مسائل زمانبندی کلاسیک، این مساله به مکان ماشین ، x، وابسته است. بخصوص اینکه زمان های آماده سازی قسمتی از داده های

ورودی نیستند. بلکه از انتخاب مکان ماشین ، x، نتیجه می شوند. بنابراین مساله زمانبندی- مکانیابی می تواند به صورت زیر نوشته شود.

$$\min_{x\in\chi} C_{\max} = \max_{j=1,\dots,n} C_j(x) \qquad (۳-۶)$$

در بخش بعدی چگونگی حل ساده ترین مساله زمانبندی- مکانیابی که شامل دو کار A_1 و A_2 و یک یال گراف $G = \{\{v_1, v_2\}, \{v_1, v_2\}\}$ می باشد را بررسی می کنیم. اطلا عات بدست آمده از حل این مساله در بخش های بعدی برای حل بکار می رود. الگوریتم هایی نیز برای حل مساله ارائه شده است.

۶-۲ زمانبندی- مکانیابی با دو کار

دو کار A_1 و A_2 با زمانهای پردازش p_1 و p_2 داده شده است که این کارها را به ترتیب با رئوس v_1 و v_2 نشان می دهیم. فاصله بین مکان های دو کار را با رابطه زیر نشان می دهیم.

$$D = d(v_1, v_2) \qquad (۴-۶)$$

فاصله نقطه $x_\alpha := ([v_1, v_2], \alpha)$، از v_1 با:

$$d(v_1, x_\alpha) = \alpha D \qquad (۵-۶)$$

و از v_2 با:

$$d(v_2, x_\alpha) = (1-\alpha) D \qquad (۶-۶)$$

نشان داده می شود. که در آن $\alpha \in [0,1]$ می باشد.
برای مکان $x_{0.5}$ خواهیم داشت:

$$r_1(x_{0.5}) = d(v_1, x_{0.5}) = d(x_{0.5}, v_2) = r_2(x_{0.5}) \qquad (۷-۶)$$

یعنی به ازای $x \in (v_1, x_{0.5})$ ، $r_1(x) < r_2(x)$ و به ازای $x \in (x_{0.5}, v_2)$، $r_2(x) < r_1(x)$.
بنابراین قانون ERD، دو منطقه زیر را روی یال تعریف می کند.

$$[v_1, x_{0.5}] \quad , \quad [x_{0.5}, v_2] \qquad (۸-۶)$$

کـه دنبالـه $\pi=\{1,2\}$ در منطقـه $[v_1, x_{0.5}]$ و دنبالـه $\pi=\{2,1\}$ در منطقـه $[x_{0.5}, v_2]$ ، جهـت پردازش بر روی ماشین، ثابت می باشند. چنین مناطقی که دنبالـه کارهـا در آن تغییـر نمـی کنـد **مناطق وبر مرتب**[۹۸] (OWR) نامیده می شود [۴۶]. مقدار تابع $Z(x)$، معادل با زمان کامل شـدن دومین کار است، یعنی;

$$Z(x)=\max\{C_1(x), C_2(x)\}=C_{\pi(2)}(x) \qquad (۹-۶)$$

حالت اول: $\pi(1)=1$, $\pi(2)=2$، یعنی ماشین در مکان $x_\alpha \in [v_1, x_{0.5}]$ قرار مـی گیـرد و ابتـدا کار A_1 پردازش می شود. در این حالت مقدار تابع هدف به صورت زیر می باشد:

$$\begin{aligned} Z(x)=C_2(x)&=\max\{C_1(x)+p_2\ , r_2(x)+p_2\} \\ &=\max\{r_1(x)+p_1+p_2\ , r_2(x)+p_2\} \end{aligned} \qquad (۱۰-۶)$$

چون پردازش کار A_2 نمی تواند قبل از کامل شدن A_1 شروع گردد، مطمئنا زمان آماده سـازی اش هم نمی تواند قبل از کار A_2 باشد. چون به ازای $\alpha \in [0\ ,0.5]$، $r_1(x_\alpha)=\alpha D$ صـعودی اسـت و $r_2(x_\alpha)=(1-\alpha)D$ نزولی است، مقدار مینیمال $Z(x_\alpha)$ برای α در نقطه تقاطع $\alpha D+p_1+p_2$ و $(1-\alpha)D+p_2$ به دست می آید.

لم ۶-۱ ([۲۸]): برای دنباله $\pi(1)=1$, $\pi(2)=2$ ، تابع هدف مقدار مینیمم خـود را اختیـار مـی کند، اگر و تنها اگر:

$$r_1(x_\alpha)+p_1=C_1(x_\alpha)=r_2(x_\alpha)$$

یعنی مکان بهینه x_α باید به گونه ای انتخاب گردد که:

$$\alpha=\alpha_{12}=\frac{D-p_1}{2D}$$

مقدار بهینه تابع هدف نیز به صورت زیر خواهد بود،

$$Z(x_{\alpha_{12}})=\frac{D-p_1}{2D}+p_2 \qquad (۱۱-۶)$$

حالت دوم: $\pi(1)=2$, $\pi(2)=1$، یعنی ماشین در $x_\alpha \in [x_{0.5}, v_2]$ مکانیابی می شود و ابتدا کار A_2 پردازش می گردد. مشابه حالت اول برای x_α خواهیم داشت:

[98] Ordered Weber regions

$$Z(x_\alpha)=C_1(x_\alpha)=\max\{C_2(x_\alpha)+p_1\ ,\ r_1(x)+p_1\}$$
$$=\max\{(1-\alpha)D+p_2+p_1\ ,\ \alpha D+p_1\} \qquad (۱۲-۶)$$

لم ۶-۲([۲۸]): برای دنباله کاری $\pi(1)=2\ ,\ \pi(2)=1$، مقداربهینه تابع هدف بـه صـورت زیـر است:

$$Z(x_{\alpha_{21}})=\frac{D+p_2}{2}+p_1 \qquad (۱۳-۶)$$

که در آن

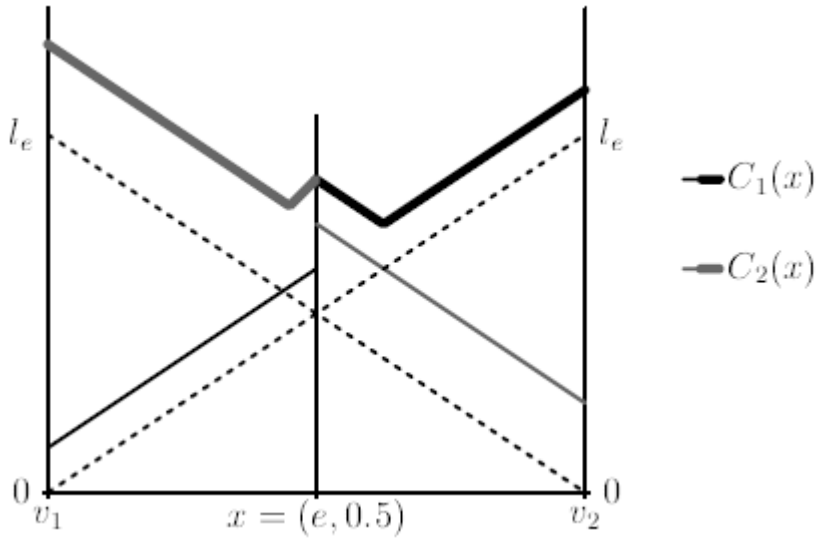
$$.\alpha_{12}=\frac{D+p_1}{2D}$$

با مقایسه $Z(x_{\alpha_{12}})$ و $Z(x_{\alpha_{21}})$ ، یک رابطه کلی برای جواب بهینه دو کار به دست می آوریم.

لم ۶-۳([۲۸]): مکان بهینه x_{α^*} برای ماشین، در مساله زمانبندی- مکانیابی تک ماشـینی بـا دو کار، به صورت زیر است:

$$\alpha^*=\begin{cases}\dfrac{D-p_1}{2D} & if \quad p_1\geq p_2\\ \dfrac{D+p_2}{2D} & if \quad p_1\leq p_2\end{cases} \qquad (۱۴-۶)$$

تجسم نتایج این بخش در شکل (۶-۳) ([۲۸]) نشان داده شده است.



شکل ۶-۳: تجسم زمان های آماده سازی (خطوط نقطه چین) و زمان های تکمیل کارها (خطوط پررنگ) و $C_{\max}$ (خطوط ضخیم) وابسته به x

۶-۳ زمانبندی- مکانیابی با بیش از دو کار

اگر ماشین قراراست روی گراف قرار بگیرد، دو فرض متفاوت برای آن وجود دارد. مکانیابی تنها روی راس ها امکان پذیر است یا اینکه ماشین روی یال ها نیز می تواند قرار بگیرد، یعنی مکانیابی روی کل گراف امکان پذیر است. برای فرض دوم، ابتدا حالت خاصی که گراف یک درخت است در نظر گرفته می شود و سپس نتایج ، به گراف های دلخواه تعمیم داده می شود.

۶-۳-۱ حالت راسی زمانبندی- مکانیابی

این مساله می تواند با شماره گذاری کل راس ها حل گردد. هر راسی ممکن است به عنوان مکان ماشین در نظر گرفته شود. بعد از ثابت شدن مکان ماشین، زمان های آماده سازی را محاسبه می کنیم و سپس مساله زمانبندی را در این راس حل می کنیم. بهترین جواب مساله زمانبندی در بین کل راس ها، مکان بهینه راسی ماشین را ارائه می دهد.
چون پیچیدگی زمانی برای قسمت زمانبندی در هر راس $O(n \log n)$ می باشد، بنابراین پیچیدگی زمانی کلی $O(n^2 \log n)$ می باشد. توجه کنید که در گراف های عمومی، این پیچیدگی با پیچیدگی $O(n^3)$ که برای محاسبه فاصله های d_{ij} با الگوریتم کوتاهترین مسیر بین هر دو جفت راس می باشد، غالب می شود ([۴۵] را ببینید).
برای اینکه محاسبات سریعتر گردد، از دو نوع کران پایین، یکی کران پایین کلی و دیگری برای مکان ثابت در راس، استفاده می شود. کران پایین کلی از مجموع زمان های پردازش به دست می آید. یعنی،

$$LB_1 := \sum_{i=1}^{n} p_i \qquad (۱۵-۶)$$

اگر مقدار تابع هدف در راسی با این کران پایین برابر گردد، می توان الگوریتم را متوقف کرد. این ملاحظات کمک می کند تا مکان بهینه ماشین را در حالت خاص به راحتی پیدا کنیم.

لم ۶-۴ ([۲۸]): اگر کار A_i، با $p_i \geq d(v_i, v_j)$ به ازای هر $j \in \{1, \cdots, n\}$ ، وجود داشته باشد، آنگاه v_i مکان بهینه ماشین می باشد.

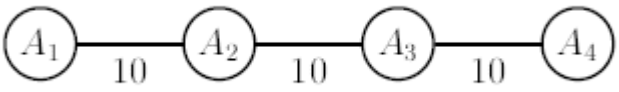
اثبات: اگر به ازای هر $j \in \{1,\cdots,n\}$ ، $p_i \geq d(v_i, v_j)$ ، آنگاه $p_i \geq r_j(v_i)$ و بنابراین $C_{\max} = \sum_{j=1}^{n} p_j = LB_1$ و چون $r_i(v_i) = 0$ ، تمام کارها می توانند بعد از A_i بدون وقفه پردازش گردند.

کران پایین موضعی، از فاصله بین راس v_j و مکان ماشین به اضافه زمان پردازش کار متناظر به دست می آید. یعنی،

$$LB_i = \max_{j=1}^{n} \{d(v_i, v_j) + p_j\} \qquad (۱۶-۶)$$

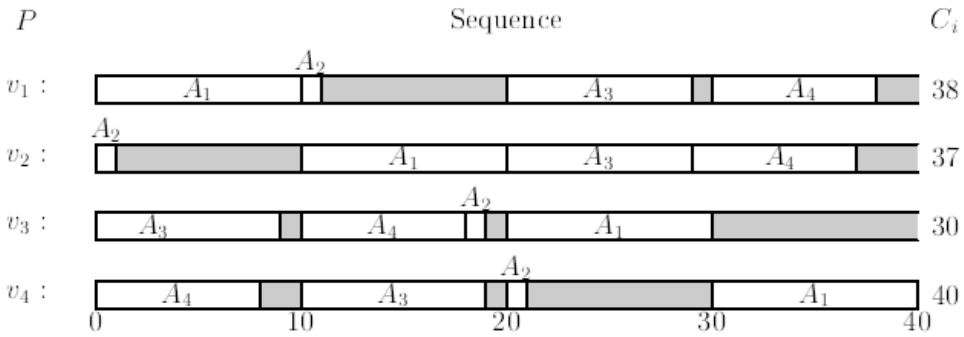
اگر LB_i برای راس v_i، بزرگتر از مقدار بهترین جواب به دست آمده تاکنون باشد، می توان محاسبه دنباله در این راس را انجام نداد.
به موجب اولین کران پایین، مرتب کردن کارها در ترتیب غیر نزولی از زمان پردازششان (قانون LPT) و جستجوی مکان بهینه ماشین در این ترتیب، هنگامی که زمان های پردازش در مقایسه با فاصله بین رئوس بالا هستند، نتایج خوبی را برای گراف های فشرده مثل گراف کامل، به بار می آورد. اگر گراف فشرده نباشد، به خصوص اگر درخت باشد استراتژی دیگری برای جستجوی مکان بهینه ماشین انتخاب می شود. در این استراتژی، کارها در ترتیب غیر نزولی از مقدار LB_i مرتب می شوند. این استراتژی نیز اگر گراف فشرده باشد ولی زمان پردازش نسبت به فاصله ها کم باشد نتایج خوبی به بار می آورد. مثال (۶-۱) نشان می دهد که نه $\arg\max_{v_i \in V} \{p_i\}$ و نه $\arg\min_{v_i \in V} \{LB_i\}$ مکان بهینه ماشین را ارائه می دهد.

مثال ۶-۱: فرض کنید $n = 4$ و $p_1 = 10$ و $p_2 = 1$ و $p_3 = 9$ و $p_4 = 8$ و فرض کنید $G = (V, E)$ گراف داده شده در شکل (۶-۴) باشد.



شکل ۶-۴: گراف مثال(۶-۱)

داریم
$\arg\max_{v_i \in V} \{p_i\} = v_1$ و $\arg\min_{v_i \in V} \{LB_i\} = v_2$ اما مکان ماشین v_3 می باشد. (شکل ۶- ۵ ([۲۸]) را ببینید)



شکل ۶-۵: نمودار گانت برای مکان ماشین در راس های مثال ۶-۱

۶-۳-۲ مسائل زمانبندی- مکانیابی مطلق در درخت ها

فرض کنید گراف درخت بدون جهت $G=(V,E)$ داده شده است و مکانیابی ماشین هر جای یال ها نیز امکان پذیر باشد. می توان از مفهوم مناطق مرتب وبر OWR ([۴۶] را ببینید) برای حل این مساله استفاده کرد. یک زیر مجموعه $R\subseteq X$ از نقاط، مناطق مرتب وبر(OWR) نامیده می شود اگر:

$$d(x,v_i)\le d(x,v_j) \iff d(y,v_i)\le d(y,v_j) \quad \forall x,y\in R,\ v_i,v_j\in V \qquad (۱۷-۶)$$

از این رو تمام فاصله های $\{d(x,v_i): v_i\in V\}$ برای تمام $x\in R$ ، به همان روش مرتب می شوند. در نتیجه دنباله کاری بهینه، برای تمام $x\in R$ یکسان است، زیرا آن دنباله، با قانون زود ترین زمان آماده سازی تعریف می شود و به ازای هر $v_i\in V$ ، $r_i(x)=d(x,v_i)$. برای ساده کردن بحث های بعدی، بدون از دست دادن کلیت فرض می شود که OWR زیر مجموعه ای از تک یال e است. یعنی به ازای بعضی $0\le \underline{\alpha}\le \overline{\alpha}\le 1$ ، $R=\{(e,\alpha): \underline{\alpha}\le\alpha\le\overline{\alpha}\}$. (اگر اینگونه نباشد، یعنی راسی درون یک OWR قرار بگیرد، آن را به چندین قسمت مجزا توسط این راس تقسیم می کنند.)

برای اینکه در یک منطقه مرتب R، ترتیب قابل تغییر نیست، به راحتی می توان مکان بهینه ماشین را در R محاسبه کرد. زمان تکمیل پردازش کل کارها معادل است با زمان کامل شدن پردازش آخرین کار. یعنی،

$$Z(\pi)=C_{\pi(n)}(x)=\max\{C_{\pi(n-1)}(x),\ r_{\pi(n)}(x)\}+p_{\pi(n)} \qquad \forall\ x\in(e,\alpha)\in R \qquad (۱۸-۶)$$

با این روش می توان زمان تکمیل شدن تمام کارها را با استفاده از رابطه زیر به دست آورد:

$$C_{\pi(i)}(x) = \max\{C_{\pi(i-1)}(x),\, r_{\pi(i)}(x)\} + p_{\pi(i)} \qquad \forall i = 1,\ldots,n\,, \quad \forall\, x \in R \qquad (۱۹-۶)$$

که

$$C_{\pi(1)} = r_{\pi(1)} + p_{\pi(1)}$$

چون G یک درخت است هر دو نقطه با یک مسیر یکتا به هم متصل می شوند. از ایـن رو بـه ازای $x \in (e,\alpha)$ با شیب ± 1 ، $r_i(x) = d(x,v_i)$ تعریف شده با رابطه (۸-۲)، در α خطی است.

نتایج و پیشنهادات

در این پایان نامه مساله زمانبندی و مکانیابی تک ماشینی مورد مطالعه و بررسی قرار گرفت که هدف اصلی این پایان نامه بود. به تبعیت از آن مسائل زمانبندی تک ماشینی و مکانیابی تک وسیله ای بر روی صفحه و خط مورد بررسی قرار گرفت. مساله زمانبندی و مکانیابی مدل تک ماشینی به قسمی که کارها بر روی صفحه و ماشین روی یک خط مشخص قرار گیرد، مساله ای جدید بود که مطرح گردید و الگوریتمی برای حل آن ارائه شد. همچنین مساله مکانیابی تک وسیله ای نیز به قسمی که مکان جدید بر روی خط مشخصی قرار گیرد، طرح گردید و مورد مطالعه قرار گرفت.

در مطالعات بعدی می توان مساله زمانبندی و مکانیابی تک ماشینی در فضای سه بعدی را مورد مطالعه قرار داد. همچنین می توان این مساله را با نرم های تر کیبی و بلو کی مورد بررسی قرار داد. می توان مساله زمانبندی و مکانیابی دو ماشینی و یا چند ماشینی را نیز بررسی کرد.

فهرست منابع و ماخذ

[1] فتحعلی ج، (۱۳۷۸)، پایان نامه کارشناسی ارشد: ”زمانبندی اجرای کارهای شرطی روی پردازشگرهای نا وابسته“ ، دانشکده ریاضی، **دانشگاه صنعتی امیرکبیر**.

[۲] جمالیان ع، (۱۳۸۸)، پایان نامه کارشناسی ارشد:” مکانیابی با جاذبه و دافعه“، دانشکده ریاضی، دانشگاه صنعتی شاهرود.

[3] O’Brien J. J., (1969), “**Scheduling handbook**”, Macgrow. Hill.

[4] Jackson J. R., (1955), “Scheduling a production to minimize maximum tardiness”, Research Report 43, **Management Science Research Project**, University of California at Los Angeles.

[5] Smith E., (1956), “ Various optimizers for single-stage production”, **Research Logistics Quarterly**, 3:59-66.

[6] Lawler E. L., (1973), “Optimal sequencing of a single machine subject to precedence constraints”, **Management Science**, 19:544–546.

[7] Karger, D., C. Stein and J. Wein, (1997), “**Scheduling Algorithms**” , A Chapter Written for the CRC Handbook on Algorithms, Boca Raton.

[8] Philips, C., C. Stein and J. Wein, (1998), “ Minimising average completion time in the presence of release dates**”, Mathematical Programming**, 199-223.

[9] Chekuri, C., R. Motwani, B. Natarajan, and C. Stein, (2001), “Approximation techniques for average completion time scheduling**”, SIAM J. Comput**., 31:146-166.

[10] Oyetunji, E.O. and A.E. Oluleye, (2007), “ Heuristics for minimizing total completion time on single machine with release time **”, Adv. Mater. Res.**, 18-19: 347-352.

[11] Oyetunji E.O., Masahudu M., (2009), “An Improved Heuristic for Minimizing Total Completion Time on Single Machine with Release Time**”, Research Journal of Mathematics and Statistics** 1(2): 65-68.

[12] Weber A. (1929), " **Uper den standort der industrient. Tubingen"**, (1909), English Trans. Theory of Location of industries, (G. J. Friedrish; Ed. And trant), Chicago university press, Chicago, Illinois.

[13] Hakimi S. L., "optimum location of switching centers and medians of a graph**"**, Operation Research, 12, pp 450-459.

[14] Love R. F., Morris J. G., Wesolowsky G. O., "**Facilities Location, models& methods**", Springer Publi

[15] Sylvester J.J., (1857), "A Question in the Geometry of Situation", Quart. **J. Math**. 1,79.

[16] Sylvester J.J., (1860), "**On Poncelet's Approximate Valuation of Surd Forms**", Philosophical Mag. XX, Fourth Series, 203-222.

[17] Jack Elzinga. Donald Hearn W.,"Geometrical Solutions for Some Minimax Location Problems", 380-394.

[18] Francis R. L., (submitted for publication, 1971), "**A Geometrical Solution Procedure for a rectilinear Minimax Location Problem**".

[19] Horst R., (1986), "A general class of branch and bound method in global optimization with some new approaches for concave minimization", **optimization Theory and Application**, 51:271-291.

[20] Tuy H., Horst R., (1988), "Convergence and restart in branch and bound algorithms for global optimization and application to concave minimization and D.C. optimization", **Mathematical Programming**, 41,161-183.

[21] Drezner Z., (2004), "The big triangle small triangle method for the solution of no convex facility location problems", **Operations research**, INFORMS, 52,1,128-135.

[22] Okabe A., Boots B., Sugihara K., (1992), "**Spatial Tessellations. Concepts and applications of Voronoi diagrams**", John Wiley & Sons, Chichester.

[23] Fortune S., (1987), "A sweep line algorithm for Voronoi diagrams", **Algorithmica** , 2,153-174.

[24] Preparata F. P., Shamos M. I., (1985), "**Computational Geometry - An Introduction**", Springer Verlag-Berlin Heideberg.

[25] Ohya T.,Iri M. and Murota T., (1984), "Improvements of the incremental of method for the voronoi diagram with computational comparisons of various algorithms", **Operations Research Society of Japan**, 277,306-336.

[26] De Berg M., Cheong O., Van Kreveld M., and Overmars M., (2008), "**Computational geometry, algorithm and applications**", Springer-Verlag.

[27] Bader D. A., Sreshta S., (2003), "**A new parallel algorithm for planarity testing**", UNM-ECE Technical Report 03-002.

[28] Hamacher HW., Hennes H., (2002), "Integrated scheduling and location models: single machine makespan problems", Technical Report, University of Kaiserslautern, Department of Mathematics, Report in Wirtschaftsmathematik Nr. 82, Kaiserslautern.

[29] Hamacher HW., Hennes H., (2007), " Integrated scheduling and location models: single machine makespan problems", **Studies in Locational Analysis**;16:77–90.

[30] Hennes H., (2005), " **Integration of scheduling and location models**", Aachen: Shaker;.

[31] Elvikis D., Hamacher H.W., Kalsch MT., (2007), "Scheduling and location (ScheLoc): makespan problem with variable release dates ", Technical Report, University of Kaiserslautern, Department of Mathematics, Report in Wirtschaftsmathematik Nr. 106, Kaiserslautern.

[32] Elvikis D., Hamacher HW., Kalsch MT., (2008), "Simultaneous scheduling and location (ScheLoc): the planar ScheLoc makespan problem ", **Journal of Scheduling**, doi: 10.1007/s10951-008-0094-4 .

[33] Marcel T. Kalsch , Zvi Drezner, (2010), "Solving scheduling and location problems in the plane simultaneously", **Computers& Operations Research**, 37 256 – 264.

[34] Blazewicz J, Ecker K, Pesch E, Schmidt G, Weglarz J., (2007). Handbook on "**scheduling: models and methods for advanced planning**", In: International handbooks on information systems. Secaucus, NJ, USA: Springer, New York, Inc.;.

[35] Brucker P.,(2007), "**Scheduling algorithms** ", Berlin: Springer;.

[36] Lawler EL., (1973), "Optimal sequencing of a single machine subject to precedence constraints ", **Management Science**, 19(5):544–6.

[37] Elvikis D., Hamacher HW., Kalsch MT.,(2008), "Simultaneous scheduling and location (ScheLoc): the planar ScheLoc makespan problem", **Journal of Scheduling**, doi: 10.1007/s10951-008-0094-4.

[38] Drezner Z., (1991), "The weighted minimax location problem with set-up costs and extensions", RAIRO—**Operations Research**;25:55–64.

[39] Michelot C., Plastria F., (2002), "An extended multifacility minimax location problem", **revisited. Annals of Operations Research**;111:167–79.

[40] Baptiste P., Brucker P., (2004), "**Scheduling equal processing time jobs**", In: Leung JY-T, editor. Handbook of scheduling: algorithms, models, and performance analysis. New York: Chapman & Hall/CRC. p. 14.1–14.37.

[41] Drezner Z., Suzuki A., (2004), "The big triangle small triangle method for the solution of nonconvex facility location problems**", Operations Research**,52(1):128–35.

[42] Lee DT., Schachter BJ., (1980), "Two algorithms for constructing a Delaunay triangulation", International **Journal of Parallel Programming**;9(3):219–42.

[43] Wendell RE., Hurter AP., (1973), " Location theory, dominance and convexity" **Operations Research**;21:314–20.

[44] Drezner Z., Goldman AJ., (1991), "On the set of optimal points to the weber problem**", Transportation Science**;25(1):3–8.

[45] Ahuja R.K., Magnanti T.L., Orlin and J.B., (1993), "**Network Flows - Theory, Algorithms, and Applications** ", Prentice Hall, Englewood Cli_s.New Jersey, .

[46] Nickel S. and Puerto J., (1999), "A uni_ed approach to network location problems", **Networks**, 34(4):283{290,.


بسم الله الرحمن الرحيم
الحمد لله رب العالمين الرحمن الرحيم مالك يوم الدين
اهدنا الصراط المستقيم صراط الذين
انعمت عليهم غير المغضوب عليهم
ولا الضالين