NEW MIXED INTEGER-PROGRAMMING MODEL FOR THE PICKUP-AND-DELIVERY PROBLEM WITH TRANSSHIPMENT

مباحث ویژه مسیر یابی

دکتر مصطفی ستاک

مازیار دهقانی

مقدمه

- در مسئله مسیریابی وسیله نقلیه با برداشت و تحویل ناوگان ناهمگنی از وسایل نقلیه مستقر در چند پایانه باید مجموعه ای از نیازها یا درخواست های حمل ونقلی را برآورده سازند.

- حمل و نقل درخواست شده میتواند شامل کالا یا افراد باشد و توابع هدف معمولا کمینه کردن هزینه های سیستم میباشد.
- مسئله VRPPD با پنجره زمانی VRPPDTW حالت خاصی از VRPTW میباشد.

- در نسخه با برداشت و تحویل VRPTW حالت خاصی از VRPPDTW است که در آن مقصدها و مبدا ها همگی در یک انبار مشترک قرار دارند.

- مدل VRPPDTWدر عمل کاربردهای فراوانی دارد از جمله جابجایی معلولین و سالمندان ، حمل دریایی و هوایی محموله ها و یگان ها ، حمل و نقل شبانه و خدمات شهری و ...

تعریف مسئله

- یک مسئله vrp ساده شامل یک دپو و چندین مشتری است کامیون ها بدون در نظر گرفتن زمان و پنجره زمانی سفارش ها را به مشتری رسانده و به دپو باز میگردند.

- در این مسئله pick up & delivery مورد بحث قرار میگیرد که علاوه بر سفارش ها که از دپو خارج میشود سفارشات بین مشتریان (گره ها) هم در نظر گرفته میشود و هر مشتری در خواست های خود را به کامیون ها میدهند و انتقال صورت میگیرد. (گراف ها در نظر گرفته میشود).

- اینگونه مسائل از دسته مسائل Np-Hard میباشد و با محدودیت هایی که در نظر میگیرند ، هزینه ها بهینه شده و جواب بهینه بدست می آید.

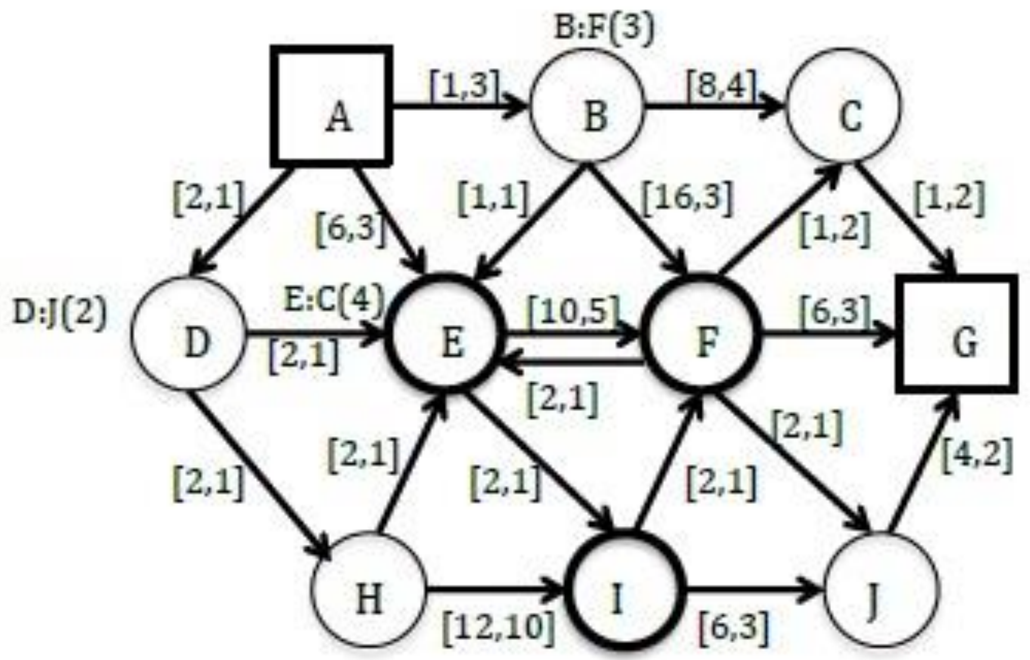

Figure 1: An example with 8 customers and 3 requests.

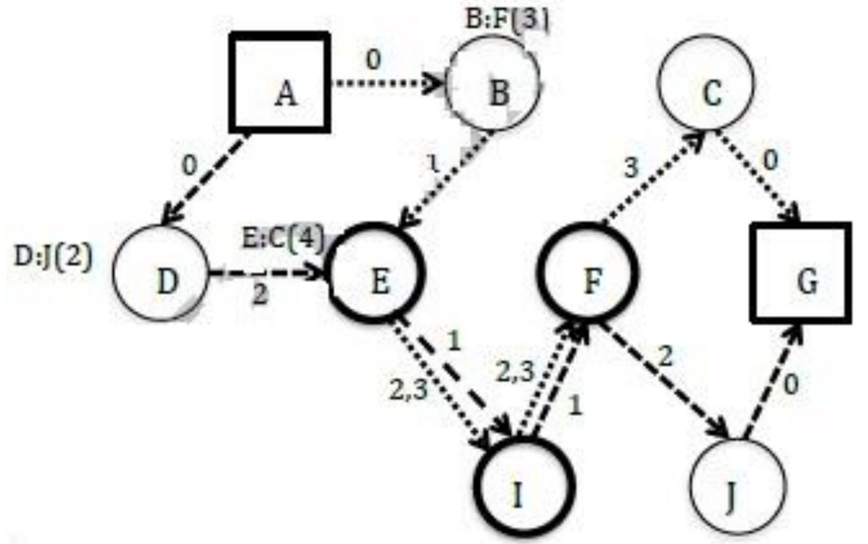

Figure 2: Optimal solution with transshipment at nodes E and F.

مدل بندی (متغیرها):

- X سفر ها
- Y سفارش مشتری
- K وسیله نقلیه
- i , j گره یا مشتری
- L رابطه بین گره ها
- r درخواست مشتری

q_r مقدار درخواست سفارش

T نقاط سفارشی

O(K) انبار اولیه

$\acute{O}$ (K) انبار نهایی

- x_{ijk} سفر از گره i به j با وسیله k
- c_{ijk} هزینه کل جابجایی از گره i به j
- هر درخواست مشتری شامل دو بخش P(r)و D(r) میباشد که شامل دریافت و تحویل از مشتری میباشد.

تابع هدف

- Min کردن هزینه جابجایی از i به j

$$Minimize \sum_{k \in K} \sum_{ij \in A} c_{ij}^{k} x_{ij}^{k}$$

- a P(r) کمترین زمان ممکن جهت دریافت از مشتری
- b P(r) بیشترین زمان ممکن جهت تحویل به مشتری

PICK UP & DELIVERY WITH OUT TIME WINDOW

- 1-یک وسیله نقلیه فقط به یکجا میتواند برود و بطور همزمان نمیتواند در دوجا قرار داشته باشد

$$\sum_{j:ij\in A} x_{ij}^{k} \leq 1 \;\; \forall k \in K, \forall i = o(k)$$

- 2-تعداد وسیله نقلیه که به یک گره وارد میشود باید از آن خارج شوند .

$$\sum_{j:ij\in A} x_{ij}^{k} = \sum_{j:jl\in A} x_{jl}^{k} \;\; \forall k \in K, \forall i = o(k), \forall l = o'(k)$$

- 3-انبار نباید وجود داشته باشد یعنی وسیله نقلیه که به یک گره وارد میشود باید از آن خارج شوند (مسیر رفت و برگشت مهم نیست)

$$\sum_{j:ij\in A} x_{ij}^{k} - \sum_{j:ji\in A} x_{ji}^{k} = 0 \;\; \forall k \in K, \forall i \in N \backslash \{o(k), o'(k)\}$$

4و5-هر درخواست (چه دریافت و چه تحویل) باید توسط یک کامیون صورت گیرد.

$$\sum_{k\in K}\sum_{j:ij\in A} y_{ij}^{kr} = 1 \;\; \forall r \in R, \forall i = p(r)$$

$$\sum_{k\in K}\sum_{j:ji\in A} y_{ji}^{kr} = 1 \;\; \forall r \in R, \forall i = d(r)$$

- 6-همه سفارشها باید دریافت و تحویل داشته باشند یعنی سفارشی گم نشود این محدودیت در مورد نقاط transshipment است.

$$\sum_{k\in K}\sum_{j:ij\in A} y_{ij}^{kr} - \sum_{k\in K}\sum_{j:ji\in A} y_{ji}^{kr} = 0 \;\; \forall r \in R, \forall i \in T$$

- 7-سفارشات چه بصورت رفت باشد چه برگشت فرقی ندارد البته این مورد در مورد نقاط غیر از نقاط transshipment کاربرد دارد (سفارشی گم نشود)

$$\sum_{j:ij\in A} y_{ij}^{kr} - \sum_{j:ji\in A} y_{ji}^{kr} = 0 \;\; \forall k \in K, \forall r \in R, \forall i \in N\setminus T$$

- 8-درخواست ها میتواند انجام شود یا نشود (در سفر از A به B سفر باید انجام شود الزامی نیست که سفارش مشتری همراهت باشد)

$$y_{ij}^{kr} \leq x_{ij}^{k} \quad \forall ij \in A, \forall k \in K, \forall r \in R$$

9-کامیون ها بر اساس ظرفیت هایشان سفارش میبرند یعنی حداکثر مقدار سفارش باید کمتر از ظرفیت کامیونها باشند.

$$\sum_{r \in R} q_r y_{ij}^{kr} \leq u_k x_{ij}^{k} \quad \forall ij \in A, \forall k \in K$$

- 10 و 11-به دلیل برنامه ریزی عدد صحیح درخواست ها یا صفر است یا یک. این 11 محدودیت در مورد زمانی است که پنجره زمانی اهمیت نداشته باشد.

$$x_{ij}^{k} \in \{0,1\} \quad \forall ij \in A, \forall k \in K$$

$$y_{ij}^{kr} \in \{0,1\} \quad \forall ij \in A, \forall k \in K, \forall r \in R.$$

- در مدلی که داریم برای حذف زیرتورها با توجه به اینکه اگر نود I زودتر از j بیاید 1= z_{ij} و در غیر اینصورت مساوی صفر است (z ها از جنس سفرند) محدودیت های زیر تعریف میشوند:

$$x_{ij}^k \leq z_{ij}^k \ \forall i,j \in N, \forall k \in K, o(k) \neq i, j \neq o'(k)$$

- 13- یا z_{ij} باید مقدار داشته باشد یا z_{ji} ، تا loop ایجاد نشود.

$$z_{ij}^k + z_{ji}^k = 1 \ \forall i,j \in N, \forall k \in K, o(k) \neq i, j \neq o'(k)$$

- 14- حداکثر دوتا میتواند یک باشد اگر سه تا یک شود loop ایجاد میشود که نمیتواند جواب بهینه باشد.

$$z_{ij}^k + z_{jl}^k + z_{li}^k \leq 2 \ \forall i,j,l \in N, \forall k \in K, o(k) \neq i, j, l \neq o'(k).$$

PICK UP & DELIVERY WITH TIME WINDOW

- 15-اگر پنجره زمانی مورد اهمیت باشد یعنی مشتری در یک بازه زمانی سفارش خود را بخواهد میبایست محدودیت های جدید اضافه کرد ، این محدودیت یعنی هیچ تاخیری نداریم و سفارش به موقع به دست مشتری رسیده است و اگر صفر باشد یعنی هیچ تاخیری نداریم.

$$\bar{t}_i^k + \tau_{ij}^k - t_j^k \leq M(1 - x_{ij}^k) \ \ \forall ij \in A, \forall k \in K$$

- 16-زمانی که به j میرسیم باید کمتر از بودن آن باشد . $t_j^k \leq \bar{t}_j^k \ \ \forall j \in N, \forall k \in K.$

- 17 و 18- دریافت و تحویل باید بین کمترین و بیشترین زمان سفر انجام گیرد.

$$a_{p(r)} \leq t_{p(r)}^k, \bar{t}_{p(r)}^k \leq b_{p(r)} \ \ \forall r \in R, \forall k \in K$$
$$a_{d(r)} \leq t_{d(r)}^k, \bar{t}_{d(r)}^k \leq b_{d(r)} \ \ \forall r \in R, \forall k \in K.$$

- 19-اگر s_{jr} = 1 باشد یعنی انتقال در نقاط transshipment توسط چندین وسیله انجام میگیرد
- 20- اگر s_{jr} = 1 باشد جابجایی بدون تاخیر صورت میگیرد.

$$\sum_{j:ji\in A} y_{ji}^{kr} + \sum_{j:ij\in A} y_{ij}^{lr} \le s_{jr}^{kl} + 1 \;\; \forall r \in R, \forall i \in T, \forall k, l \in K, k \neq l$$

$$t_j^k - \bar{t}_j^l \le M(1 - s_{jr}^{kl}) \;\; \forall r \in R, \forall j \in T, \forall k, l \in K, k \neq l.$$

- در این مدل MIP که برای PDPT عمومی میباشد ما میتوانیم جنبه های مختلف آنرا کاربردی تر در نظر بگیریم مثلا میتوانیم چندین انبار نهایی برای وسیله نقلیه ، همچنین وسایل نقلیه متفاوت برای ظرفیت های متفاوت در نظر بگیریم ، همچنین بتوان حداقل وسیله مورد نیاز را برای درخواست های مشتریان اختصاص داد و در نهایت اگر مشتری بیش از یک در خواست داشت بتوان آنرا انجام داد.

- تمامی شرایط کاربردی بالا را میتوان در این مدل MIP با اضافه کردن محدودیت های مختلف مدل سازی کرد.

- Splittable pickup-and-delivery request

- درخواست های مشتریان از لحاظ تنوع مدل و مقدار میتواند غیر یکنواخت باشد که به اصطلاح بارهایی که قابلیت شکست دارند.
- جدا کردن درخواست ها یکی از روش های متداول در دنیای واقعی میباشد بعنوان مثال در جابجایی محصولات پتروشیمی ، دانه های غذایی و معدنی و ...

$$y_{ij}^{kr} \geq 0 \;\; \forall ij \in A, \forall k \in K, \forall r \in R.$$

- Vehicle depots and routes

- در حرکت وسایل نقلیه از انبارها دو حالت میتواند اتفاق بیافتد 1- مسیرهای وسیله نقلیه محدود شده باشد (در فاصله خاصی بماند) 2- فقط یک وسیله خاص میتواند برود
- حداکثر فاصله وسیله نقلیه k از مسیر مجاز: l^k

$$\sum_{ij \in A} l_{ij} x_{ij}^k \leq L^k.$$

- Number and types of vehicles :
  - اگر محدودیت شماره 1 را بجای >= علامت = قرار دهید بدین منظور است که کامیون مطابق ظرفیتش میتواند در یکجا باشد.
  - اگر برای یک سفارش خاص مشتری متقاضی نوع خاص از وسیله نقلیه باشد به تعداد آن وسیله نقلیه هزینه ثابت آن وسیله (f^k) را شامل میشود.

$$\sum_{k\in K}\sum_{ij\in A, i\in o(k)} f^k x_{ij}^k + \sum_{k\in K}\sum_{ij\in A} c_{ij}^k x_{ij}^k.$$

  - بعضی از سفارشات ناهمگن و خاص میباشند مانند مواد فاسد شدنی

$$y_{ij}^{kr} = 0, \forall ij \in A, \forall k \in K, \forall r \in R$$

- # Vehicle stops:

- وسیله نقلیه میتواند اجازه داشته باشد که انتهای مسیر خود را در نقاط delivery و transshipment به جای نقاط انتهایی دپوی تخصیص داده شده توسط مسئله قرار دهد همچنین در نقاط transshipment بایستد تا درخواست جدیدی بگیرد.

$$\sum_{j:ij\in A} x_{ij}^{k} - \sum_{j:ji\in A} x_{ji}^{k} \leq 0 \quad \forall k \in K, \forall i \in T \cup D \setminus \{o(k), o'(k)\}$$

$$\sum_{k\in K}\sum_{j:ij\in A} y_{ij}^{kr} - \sum_{k\in K}\sum_{j:ji\in A} y_{ji}^{kr} \leq 0 \quad \forall r \in R, \forall i \in T \cup D.$$

- # Transshipment:

- تعداد دفعاتی که وسیله نقلیه میتواند transshipment را انجام دهد متفاوت است (بعضی از وسایل نقلیه اصلا انجام نمیدهند ، بعضی ها یکبار و بعضی ها در نقاطی خاص بیشتر از دوبار انجام نمیدهند.

$$\sum_{r\in R}\sum_{l\in K, l\neq k} s_{jr}^{kl} \leq 1 \quad \forall j \in T, \forall k \in K.$$

$$-\sum_{r\in R}\sum_{l\in K, l\neq k} s_{jr}^{kl} \quad \forall k \in K.$$

- Dial-a-ride problem with transshipment :

- مسئله به این نحو تعریف میشود که مشتریان خودشان به دنبال مینیمم هزینه سفر از جایگاه اولیه به مقصد میباشند. مثلا برای دسترسی قشر سالخورده یا معلول به مراکز اورژانس یا پایانه های مسافربری از شهری به شهر دیگر. (تاکس تلفنی)
- همچنین DARPT دیرکرد ماشین ها و زمان سفر را بهینه میکند.

PDPT by setting $q_r = 1, \forall r \in R$. Compared to the DARP

- روش حل MIP میباشد والگوریتم (رویکرد حل) آن Branch-and-Cut و Branch-and-Bound میباشد.

- برای حل مسئله MIP میبایست از SIMPLEX استفاده کنیم که با الگوریتم Branch-and-Cut و Branch-and-Bound ترکیب شده و مدل MIP را میتوان حل کرد.

- در این مقاله از نرم افزار GUROBI استفاده شده است و روش حل PDP میباشد ، این نرم افزار چه بر اساس پنجره زمانی و چه بدون آن، CPU time و Optimal value را حساب میکند.

Table 1: Optimal values and CPU times for PDPT without time windows (10 nodes).

Instance	Optimal value	CPU time	Optimal value (transshipment)	CPU time (transshipment)	Gap in optimal values (%)
10n1a	1121.58	4.24	1083.09	9.57	3.43
10n2a	931.06	33.92	867.01	8.54	6.88
10n3a	1221.65	54.46	1221.65	62.15	0
10n4a	1105.11	108.79	1105.11	671.03	0
10n5a	848.89	291.84	833.89	311.48	1.77
10n6a	1018.35	1.88	1018.35	8.78	0
10n7a	1272.59	223.85	1249.51	137.92	1.81
10n8a	750.52	206.56	750.52	808.83	0
10n9a	1031.93	516.64	1028.19	1139.01	0.36
10n10a	1224.71	65.12	1224.71	94.47	0

Table 2: Optimal values and CPU times for PDPT with time windows (10 nodes).

Instance	Optimal value	CPU time	Optimal value (transshipment)	CPU time (transshipment)	Gap in optimal values (%)
10n1a	1121.58	5.11	1083.09	15.85	3.43
10n2a	936.15	26.4	933.88	24.52	0.24
10n3a	1221.65	1.59	1221.65	8.93	0
10n4a	1124.66	102.97	1123.18	283.71	0.13
10n5a	872.92	76.18	865.62	112.47	0.84
10n6a	1018.35	0.53	1018.35	0.79	0
10n7a	1249.51	8.4	1249.51	113.01	0
10n8a	750.52	35.74	750.52	82.16	0
10n9a	1031.93	64.65	1031.93	100.06	0
10n10a	1269.23	4.81	1267.58	7.51	0.13

•

Table 3: Optimal values and CPU times for PDPT with time windows (14 nodes).

Instance	Optimal value	CPU time	Optimal value (transshipment)	CPU time (transshipment)	Gap in optimal values (%)
14n1a	695.78	6027.34	683.95	11470.86	1.70
14n2a	599.57	5.04	594.72	17.86	0.81
14n3a	671.5	852.47	650.52	1161.82	3.12
14n4a	592.61	94.21	589.35	2683.32	0.55
14n5a	566.13	521.39	566.13	3997.69	0
14n6a	1021.28	2076.35	1005.24	18182.82	1.57
14n7a	1048.71	272.34	1013.79	1079.37	3.33
14n8a	819.11	1699.11	816.6	9368.72	0.31
14n9a	704.02	71.26	704.02	1354.29	0
14n10a	536.44	27.33	535.02	85.78	0.26

کار در آینده:

- 1.می توان سطح خدمت را چندتا در نظر بگیریم به عنوان مثال قیمت ، زمان تحویل ، رضایت برای مشتری که می توان از جنبه فازی و تخصیص برای رضایت مشتری استفاده کرد.

- 2. می توان در بین مسیر، دریافت و تحویل بین خود وسایل نقلیه انجام داد بطور مثال در بین مسیر 2 تا کامیون یکی کرد و یا در بین راه آن وسیله نقلیه با وسیله نقلیه کوچک جابجا کرد.

- 3. امکان سنجی انجام داد که با کدام وسیله نقلیه مقرون به صرفه تر است که تحویل انجام پذیرد بطور مثال با کشتی مقرون به صرفه تر است یا کامیون یا وانت یا هر وسیله دیگر.

- 4. به جای یک depo چندین depo در نظر بگیریم به عنوان مثال چندین depoدر سراسر کشور داشته باشیم که بتوانیم انتقال و تحویل به مشتری سریعتر و مقرون به صرفه تر انجام گیرد.و انبارها را بصورت خوشه ای و منطقه ای در نظر بگیریم و آنها را به یک انبار مرکزی اتصال دهیم که در صورت لزوم میتوان انتقال بین انبار ها را هم انجام داد.

با تشکر از حسن توجه شما