به نام خداوند بخشنده مهربان

درس بهینه سازی محدب

گزارش سمینار درس

استاد درس: دکتر هادی امیری

دانشجو: مینو احمدی

810897032

دانشکده علوم مهندسی، دانشگاه تهران

Miinouahmadii@gmail.com

فهرست مطالب

مدل بهینه سازی نوین برای انالیز سبد خرید مشتری با در نظر گرفتن تخصیص و حل مسئله با الگوریتم ژنتیک

مقدمه و معرفی کلی

امروزه تحلیل سبد بازار یکی از حوزه های تحقیقاتی مورد علاقه داده کاوی است که بیشتر مورد توجه محققان قرار گرفته است. اما اکثر تحقیقات مرتبط بر روی الگوریتم‌های سنتی و اکتشافی با عوامل محدود متمرکز شده‌اند که تنها عوامل تاثیرگذار در تحلیل بازار سبد نیستند. در این مقاله برای مدل‌سازی و تحلیل کارآمد داده‌های سبد بازار، مدل بهینه‌سازی با در نظر گرفتن پارامتر تخصیص به‌عنوان یکی از عوامل مهم و تأثیرگذار بر نرخ فروش پیشنهاد شده است. رویکرد الگوریتم ژنتیک برای حل مسئله برنامه‌ریزی باینری غیرخطی فرمول‌بندی شده و از یک مثال عددی برای نشان دادن مدل ارائه شده استفاده می‌شود. نتایج ارائه شده نشان می‌دهد که راه‌حل‌های به‌دست‌آمده واقعی‌تر و کاربردی‌تر به نظر می‌رسند.

تعداد قابل توجهی از تحقیقات در مورد تکنیک‌های کاوی قواعد ارتباطی و روش‌های بهینه‌سازی در یک تحقیق جداگانه وجود دارند، اما روش‌های بهینه‌سازی ریاضی در کنار داده کاوی و همچنین به‌کارگیری روش‌های فراابتکاری انجام نشده است.

در این مقاله علاوه بر در نظر گرفتن قواعد هم انجمنی ترکیب محصولات سبد خرید ، جایگاه هر محصول در قفسه های مختلف نیز به عنوان عامل بسیار مهم در انتخاب محصول توسط مشتری نیز در نظر گرفته شده است.(مثلا کالا هایی که در قفسه های نزدیک به در ورودی و خروجی باشند با احتمال بیشتری در مقایسه با محصولات در جاهای دیگر ، فروش میروند.

یک مدل بهینه سازی غیرخطی صفر و یک برای قوانین انجمن استخراج و قرار دادن محصولات در قفسه ها. شایان ذکر است که مدل ریاضی پیشنهادی و به‌کارگیری فراابتکاری مناسب در تحقیق قبلی مورد توجه قرار نگرفته است و ما معتقدیم که مدل پیشنهادی چارچوبی جامع برای فرمول‌بندی واقعی‌تر مسائل دنیای واقعی ارائه می‌دهد.

سازماندهی این مقاله به شرح زیر است: در بخش بعدی شرح و فرمول مسئله ارائه شده است. در بخش سوم از الگوریتم ژنتیک برای حل مدل پیشنهادی استفاده شده است. یک مثال گویا برای روشن شدن مدل پیشنهادی در بخش چهارم ارائه شده است. در نهایت، نتیجه گیری ذکر شده است.

شرح و فرمول مسئله

گزارش های داده بازار را در نظر بگیرید که شامل اقلام خریداری شده توسط مشتریان است. مدیر یک سوپرمارکت می خواهد جذابیت قرار دادن محصول در قفسه ها را به حداکثر برساند. یعنی ارزش جذابیت مربوط به قوانین انجمن استخراج شده و محل قفسه ها. منطق به حداکثر رساندن جذابیت با ملاحظات مکان مبتنی بر این واقعیت است که استخراج قوانین انجمن به حداکثر رساندن اثر فروش متقابل کمک می کند،

اما واضح است که مکان قفسه ها تأثیر غیرقابل انکاری بر نرخ فروش دارد. به عنوان مثال، محصولاتی که در نزدیکی درهای ورودی یا خروجی قرار می گیرند، شانس بیشتری برای خرید دارند. بنابراین می توان گفت که عملکرد ترجیحی مدیر سوپرمارکت به پارامترهای زیر بستگی دارد: سود حاصل از فروش، پشتیبانی و اطمینان هر جفت محصول و امکان فروش هر قفسه برای هر محصول.(کانفیدنس لول و ساپورت نامبر) این پارامترها در تابع ترجیحی زیر ادغام می شوند:

تابع هدف مسئله:

$$\sum_{i=1}^{m-1}\left[\sum_{l=i+1}^{m}\left[C_{il}+C_{li}\sum_{k=1}^{P}[b_i v_{ik}+b_l v_{lk} x_{ik} x_{lk}]\right]\right]$$

m: تعداد محصولات

p: تعداد قفسه ها

C_{il}: سطح اطمینان هر ترکیب هم نشینی محصول i در لیستی که l نیز وجود دارد.

b_i: سود فروش از محصول i

v_{ik}: احتمال فروش محصول i زمانی که در قفسه k قرار گرفته است.

x_{ik}: محصول i که در قفسه K جا دارد.

در این جا تابع هدف ما ماکزیمم سازی جذابیت فروش است که منجر به بیشینه شدن سود حاصل از فروش میشود.

محدودیت های مسئله:

محدودیت اول: محدودیت ظرفیت

$$\sum_{i=1}^{m} x_{ik} \leq U_k \quad ; k = 1,2,\dots,P$$

U_k: میزان گنجایش قفسه k ام

مجموع تمام انواع محصولات قرار گرفته در قفسه k ام محدود به ظرفیت قفسه است.

محدودیت دوم:

$$x_{ik}x_{lk}(S_{il} - S_{min}) \geq 0 \quad ; \forall\, i,l \in \{1,2,\dots,m\}$$

S_{il}: تعداد تکرار هر ترکیب (rule) در سبدهای خرید مشتریان

S_{min}: حداقل ساپورت که خودمان بسته به مسئله تعیین میکنیم

تعداد ساپورت هر rule نباید کمتر از حداقل ساپورت تعیین شده باشد.

محدودیت سوم:

$$\sum_{k=1}^{P} x_{ik} = 1 \quad ; \; i = 1,2,\dots,m$$

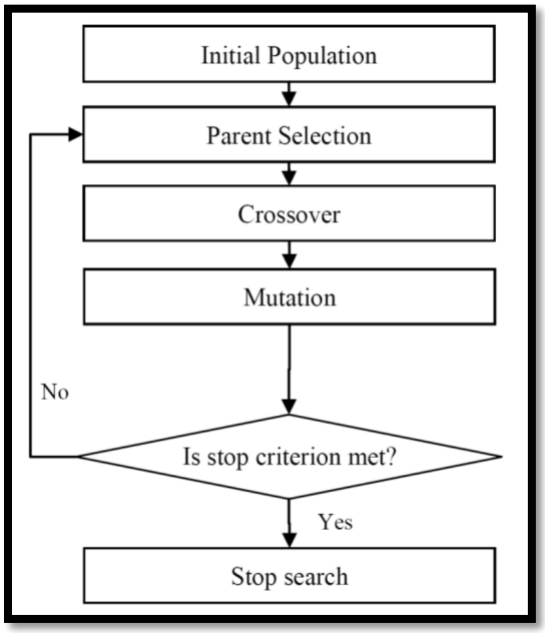
هر نوع محصول فقط میتواند در یک قفسه قرار گیرد.

محصول i ام نمیتواند هم در قفس j و هم درقفس j+m باشد.

با توجه به این واقعیت که تابع هدف و محدودیت ها توابع غیر خطی هستند که در آنها متغیرهای تصمیم باینری هستند. ما با یک فضای عملی ناهموار سروکار داریم که احتمال به دام افتادن در بهینه محلی را افزایش می دهد. علاوه بر این، می توان ثابت کرد که مدل توسعه یافته متعلق به کلاسی از مسائل سخت محاسباتی است که به آن مسائل Np-hard می گویند. بنابراین در بخش بعدی، یک رویکرد راه حل مبتنی بر GAs برای حل مدل ریاضی پیشنهادی توسعه می‌یابد.

الگوریتم ژنتیک

الگوریتم ژنتیک به دسته‌ای از روش‌های فراابتکاری معروف به روش‌های جستجوی تصادفی تعلق دارد که از انتخاب تصادفی عملگرها در استراتژی جستجوی خود استفاده می‌کنند. در این بخش، GA ها را برای به دست آوردن راه حلی برای مدل ارائه شده در بخش قبل پیاده سازی می کنیم. مکانیسم کلی GAs در شکل 1 نشان داده شده است.



مکانیزم الگوریتم ژنتیک

ثابت شده است که ویژگی های GA مانند متقاطع، جهش و عملکرد جریمه و همچنین مکانیسم های انتخاب تأثیر عمده ای بر کیفیت راه حل های ارائه شده دارد. برای اجرای GA، ما باید مفاهیم اساسی زیر را تعیین کنیم: نمایش کروموزوم، متقاطع و جهش و استراتژی های انتخاب.

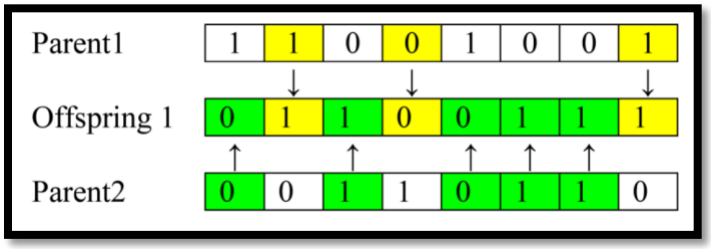
نمایی از کروموزوم و تاثیر عملگر های موجود در GA

استراتژی های انتخاب کروموزوم در نظر گرفته شده برای هر سلول نشان داده شده است که در آن مقدار هر ژن باینری است که وقتی محصول i به قفسه k تخصیص داده می شود 1 می گیرد، در غیر این صورت 0 می باشد.

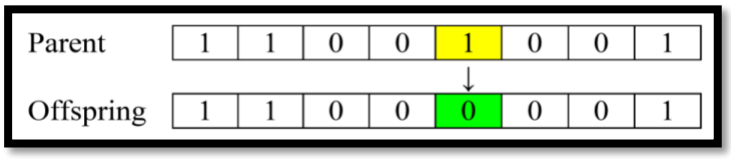
x_{11}	x_{12}	...	x_{1P}	...	x_{m1}	x_{m2}	...	x_{mP}

عملگرهای تقاطع و جهش برای کشف مناطق ناشناخته فضای امکان پذیر استفاده می شود .مجموعه ای از موقعیت ها از والد اول به طور تصادفی انتخاب می شود .این مقادیر در موقعیت های یکسان فرزندان

کپی می شوند و مقادیر باقی مانده توسط همان ژن های والد دوم برآورده می شوند .فرزندان دیگری نیز به همین ترتیب تولید می شوند.



عملگر جهش طوری است که یک موقعیت تصادفی انتخاب شده و مقدار ان فلیپ فلاپ میشود.



استراتژی های انتخابی

در اینجا ما از دو استراتژی انتخاب استفاده می کنیم، چرخ رولت برای انتخاب والدین برای تولید فرزندان و نخبه گرایی برای انتخاب نسل بعدی از فرزندان و والدین به عنوان مکانیسم انتخاب بازمانده. برای کسب اطلاعات بیشتر در مورد چرخ رولت و مکانیسم های نخبه گرایی، خوانندگان علاقه مند به سیواناندام و دیپا (2008) مراجعه می کنند.

بررسی امکان سنجی

در مسائل بهینه‌سازی با فضای امکان‌پذیر گسسته، ایجاد راه‌حل‌های امکان‌پذیر یک نگرانی عمده است. دو رویکرد برای مقابله با این مشکل وجود دارد. اولین مورد، جستجوی فضای امکان پذیر از طریق ایجاد راه حل های امکان پذیر است که روشی زمان بر است و همچنین ممکن است منجر به بهره برداری و اکتشاف مؤثر نشود. روش دیگر می تواند استفاده از تابع پنالتی باشد. به این ترتیب تمام فرزندان تولید شده اعم از امکان پذیر یا غیرقابل اجرا پذیرفته می شوند و ارزش جریمه به راه حل های غیرقابل تحقق بر اساس درجه امکان پذیری آنها تعلق می گیرد. درجه غیرممکن بر اساس نسبت محدودیت های نقض شده محاسبه می شود. بنابراین ارزش تناسب هر فرزند شامل تابع هدف و مقادیر جریمه می شود.

در بخش بعدی، یک مثال عددی گویا برای روشن شدن مدل توسعه‌یافته و رویکرد راه‌حل پیشنهادی ارائه می‌شود.

مثال عددی

در این بخش، نمونه‌ای از داده‌های سبد بازار برای توصیف مدل پیشنهادی و رویکرد راه‌حل مبتنی بر GA شبیه‌سازی می‌شود. برای این کار ده کالا در نظر گرفته شده است که باید در سه قفسه تخصیص داده شوند. بر اساس موقعیت قفسه ها، هر قفسه تأثیر متفاوتی بر امکان فروش کالاهای تخصیص یافته دارد. این احتمالات فروش می تواند توسط کارشناسان تعیین شود. این مقادیر در جدول 1 ارائه شده است.

Table 1. *The selling possibility of each goods* ($v_{ik}; i = 1, \dots, 10, k = 1,2,3$)

Goods / Shelves	1	2	3	4	5	6	7	8	9	10
1	0.8	0.6	0.1	0.9	0.1	0.1	0.5	0.1	0.2	0.1
2	0.5	0.2	0.5	0.4	0.5	0.3	0.5	0.1	0.3	0.8
3	0.9	0.5	0.4	0.1	0.9	0.7	0.5	0.7	0.8	0.1

ویژگی دیگری که در تخصیص محصولات تاثیر زیادی دارد، سود فروش است. بنابراین منطقی است که برای به حداکثر رساندن سود مورد انتظار از فروش، محصولات با مزایای بالاتر باید به قفسه هایی با امکان فروش بالاتر اختصاص داده شوند. جدول 2 مقادیر سود محصولات را نشان می دهد

Table 2. *The benefit of each product ($/unit) (* $b_i; i = 1, \dots, m$ *)*

Goods	1	2	3	4	5	6	7	8	9	10
Benefit	40	15	70	20	15	25	10	10	22	5

برای داده های شبیه سازی شده، مقادیر اطمینان و پشتیبانی به صورت جدول به ما داده شده است.

Table 3. *The confidence values for simulated data* (C_{il} *)*

Goods	1	2	3	4	5	6	7	8	9	10
1	1	0.67	0.42	0.44	0.36	0.28	0.28	0.47	0.31	0.33
2	0.51	1	0.38	0.38	0.38	0.28	0.23	0.47	0.3	0.3
3	0.33	0.39	1	0.48	0.43	0.26	0.3	0.46	0.37	0.3
4	0.36	0.4	0.49	1	0.36	0.31	0.29	0.53	0.4	0.27
5	0.36	0.5	0.56	0.44	1	0.22	0.25	0.44	0.39	0.36
6	0.28	0.36	0.33	0.39	0.22	1	0.56	0.53	0.28	0.28
7	0.32	0.35	0.45	0.42	0.29	0.65	1	0.39	0.23	0.32
8	0.35	0.45	0.43	0.49	0.33	0.39	0.24	1	0.47	0.33
9	0.35	0.45	0.55	0.58	0.45	0.32	0.23	0.74	1	0.29
10	0.4	0.47	0.47	0.4	0.43	0.33	0.33	0.53	0.3	1

برای داده های شبیه سازی شده، مقادیرپشتیبانی به صورت جدول به ما داده شده است.

Table 4. *The support values for simulated data (S_{il})*

Goods	1	2	3	4	5	6	7	8	9	10
1	0.36	0.24	0.15	0.16	0.13	0.1	0.1	0.17	0.11	0.12
2	0	0.47	0.18	0.18	0.18	0.13	0.11	0.22	0.14	0.14
3	0	0	0.46	0.22	0.2	0.12	0.14	0.21	0.17	0.14
4	0	0	0	0.45	0.16	0.14	0.13	0.24	0.18	0.12
5	0	0	0	0	0.36	0.08	0.09	0.16	0.14	0.13
6	0	0	0	0	0	0.36	0.2	0.19	0.1	0.1
7	0	0	0	0	0	0	0.31	0.12	0.07	0.1
8	0	0	0	0	0	0	0	0.49	0.23	0.16
9	0	0	0	0	0	0	0	0	0.31	0.09
10	0	0	0	0	0	0	0	0	0	0.3

گنجایش قفسه ها در جدول زیر امده است.

Table 5. *The capacity of shelf k (U_k)*

Shelves	1	2	3
Capacity	2	4	4

بر اساس اطلاعات فوق، مدیر بازار تصمیم می گیرد تا تابع ترجیحی خود را به حداکثر برساند. مسئله با استفاده از مدل بهینه سازی ریاضی توسعه یافته فرموله شده و GA توصیف شده با پارامترهای زیر برای حل مدل بهینه سازی استفاده شده است. میزان جهش و عملگرهای متقاطع به ترتیب 0.6 و 0.4 است. حجم جمعیت 500 نفر در نظر گرفته شده است و یک درصد از جمعیت شامل والدین و فرزندان به عنوان نخبگان برای انتقال مستقیم نسل بعدی انتخاب می شوند. GAs با حداکثر 100 نسل به عنوان معیار خاتمه اجرا می شود. GA با 1000 سیکل اجرا می شود و بهترین راه حل در جدول 6 به عنوان تخصیص بهینه گزارش شده است.

Table 6. *The optimum solution*

Goods / Shelves	1	2	3	4	5	6	7	8	9	10
1	0	0	0	0	0	1	0	0	0	1
2	0	0	1	0	1	0	0	1	0	0
3	1	1	0	1	0	0	1	0	0	0

برای راه حل فوق، مقدار هدف 306.9 است. به منظور شناسایی اینکه پاسخ ارائه شده توسط GA راه حل بهینه محلی یا جهانی است، در مرحله اول تجزیه و تحلیل حساسیت پارامترهای GA را انجام دادیم. برای انجام این کار، ما اندازه جمعیت بزرگ‌تر را با مکانیسم‌های مختلف تولید جمعیت اولیه، متقاطع، جهش و استراتژی‌های انتخاب برای حل مثال عددی اعمال کردیم. نتایج نشان داد که استراتژی‌های متقاطع، جهش و انتخاب پیشنهادی GA را به خوبی تنظیم می‌کند که در آن فضای امکان‌پذیر را به درستی مورد

بهره‌برداری و کاوش قرار می‌دهد. در مرحله دوم از آنجایی که مثال عددی مقیاس کوچکی داشت، مثال عددی را با استفاده از نرم افزار بهینه سازی GAMS حل کردیم. نتیجه GAMS نشان داد که راه حل بهینه بدست آمده توسط GA یک بهینه جهانی است.

بحث و نتیجه گیری

در این مقاله با در نظر گرفتن اثرات قرار دادن کالا در قفسه ها، یک مدل ریاضی جدید توسعه داده شد. برای این منظور، با در نظر گرفتن حمایت و اطمینان به عنوان دو عامل مهم تحلیل سبد بازار، موقعیت محصولات به عنوان پارامتر دیگری در نظر گرفته شد که ممکن است بر نرخ فروش تأثیر بگذارد. این عوامل و اثرات آنها از نظر تابع هدف ، به صورت یک مدل برنامه ریزی غیرخطی صفر و یک فرموله شده و با استفاده از GA حل شد. شایان ذکر است که در اکثر تحقیقات گزارش شده، از الگوریتم های فراابتکاری مانند GA برای کشف قوانین ارتباط استفاده شده است، در حالی که در مقاله حاضر، فرض بر این است که قوانین پشتیبانی و اطمینان مجموعه داده ها قبلاً وجود داشته است. شناخته شده. بنابراین، به جای تمرکز بر روی بهینه‌سازی قوانین تداعی، در این مقاله از GA برای استخراج مکان بهینه کالاها در قفسه‌هایی استفاده شد که در آن تابع هدف تصمیم‌گیرنده حداکثر شده است.

در مورد عملکرد و ویژگی‌های GA توسعه‌یافته، شایان ذکر است که از یک سو، تنظیم صحیح پارامترهای الگوریتم‌های تکاملی عمدتاً بر عملکرد الگوریتم‌ها تأثیر می‌گذارد. از سوی دیگر، تنظیم الگوریتم های تکاملی مانند GA به شدت به ویژگی های مدل بهینه سازی ریاضی بستگی دارد.

بنابراین، می توان انتظار داشت که در حالی که یک GA به خوبی تنظیم شده ممکن است با موفقیت یک راه حل بهینه جهانی را برای یک مدل ریاضی معین به دست آورد، استفاده از این GA برای یک مسئله دیگر ممکن است به یک راه حل محلی و رضایت بخش منجر شود. در نتیجه مقایسه GA ارائه شده با سایر الگوریتم‌های تکاملی توسعه‌یافته در مدل‌های مختلف ریاضی موردی نیست. با این حال، به منظور اطمینان از عملکرد GA پیشنهادی، ما راه حل بهینه به دست آمده توسط GA را با GAMS در مسائل ریاضی در مقیاس کوچک مقایسه کردیم. نتیجه ارائه شده نشان داد که GA پیشنهادی به خوبی تنظیم شده است و می تواند با موفقیت برای استخراج یک راه حل بهینه استفاده کند.

به طور خلاصه، در حالی که در مقاله حاضر تأثیر مکان یابی کالا بر فروش گنجانده شده است و راه حل واقع بینانه تری ارائه شده است، اما اشکالاتی وجود دارد که باید در تحقیقات آتی مورد توجه قرار گیرد. در این مقاله، قوانین تداعی از قبل شناخته شده فرض شده است، در حالی که می توان از روش و الگوریتم مناسب برای کشف قوانین تداعی نیز استفاده کرد. در مدل حاضر فرض بر این بود که پارامترهای مدل واضح هستند و مقادیر دقیق پارامترها مشخص است. اما همانطور که می توان انتظار داشت، به دلیل عدم قطعیت محیط، چنین فرضی معتبر نیست و توسعه مدل های ریاضی نامطمئن ممکن است به استخراج مدل های واقعی تر و به نوبه خود راه حل ها کمک کند. طبقه بندی کالاها، تعریف معیارهای جذابیت مناسب نیز جهت گیری های احتمالی تحقیقات آینده است.

در این جا مثال عددی داده شده در مقاله پیاده سازی شده است .

با اینکه کتابخانه های زیادی برای حل الگوریتم ژنتیک و فیتنس و جهش و تقاطع وجود دارند همه ی این تابع ها بدون استفاده از کتابخانه ها از اول پیاده سازی شده اند.

کلاس Scheduler برای پیاده سازی و برنامه ریزی برای تخصیص محصولات به قفسه ها.

در تابع generateInitialPopulation در ابتدا جمعیت اولیه را با ۵۰۰ کروموزوم میسازیم.

کروموزوم ها ارایه ای ۳۰ تایی از اعداد ۰و ۱ هستند (پارامتر تخصیص) به این صورت که برای هر محصول (۱۰ تا) سه قفسه وجود دارد.

در جمعیت اولیه اعداد ۰و۱ به صورت کاملا رندوم و بدون ارضا کردن هیچ محدودیتی انتخاب میشود.

در تابع schedule به تعداد ۱۰۰ بار جمعیت جدید تولید میکنیم.

در تابع generateNewPopulation سه کار اصلی انجام میگیرد:

1 - تابع crossover را روی جمعیت اعمال میکنیم.
2 - تابع mutate را روی جمعیت اعمال میکنیم.
3 - جمعیت نخبگان نسل قبل را به نسل جدید اضافه میکنیم.

در تابع caculateFitness محدودیت ها را در هر کروموزوم بررسی میکنیم.

در تابع check_column_limits محدودیت وجود هر نوع محصول فقط در یک قفسه نظر گرفته میشود.

در تابع check_shelf_limits محدودیت گنجایش قفسه ها در نظر گرفته میشود.

در تابع check_support_constraint محدودیت ساپورت چک میشود.

در تابع calculate_preference_function تابع هدف را حساب میکنیم.

خروجی تابع همان مقدار ۳۰۶ شد که در مثال نیز مقدار بهینه همین مقدار حاصل شده است.

تصویر خروجی :


TERMINAL
PROBLEMS
OUTPUT
DEBUG CONSOLE
306.0
306.0
306.0
306.0
306.0
306.0
306.0
306.0
306.0