

آموزش برنامه نویسی

GAMS

در قالب کد نویسی

تهیه و تنظیم:

علیرضا بهاری

E-Mail: ar_bahari@yahoo.com

آموزش برنامه نویسی

GAMS

(General Algebraic Modeling System)

در قالب کد نویسی

نمایش برنامه نویسی GAMS به گونه‌ای است که خواندن و درک آن برای خواننده و یا رایانه آسان است. به عبارت دیگر می‌توان گفت، برنامه GAMS توضیح محکمی در مورد آنچه به زبان GAMS نوشته شده، می‌باشد و با نگاه کردن به آن می‌توان گزاره‌های آن را تجزیه و تحلیل کرد و از توضیحاتی که قبلاً به صورت جداگانه برای این زبان نوشته می‌شد، بی‌نیاز است. علاوه بر این طراحی GAMS به گونه‌ای است که شامل ویژگی‌های اشاره شده در زیر است که نیاز استفاده کنندگان از این نرم‌افزار را برآورده می‌کند.

- در نمایش گزاره‌های نرم‌افزار از نهایت اختصار و ظرافت استفاده شده است. یعنی داده‌ها را به ساده‌ترین و ابتدایی‌ترین شکل می‌توان وارد نرم‌افزار کرد.
- همچنین تمام عملیاتی که حین اجرای برنامه روی مدل پیاده می‌شود و همچنین گزارش ریز تغییرات برای بررسی بیشتر استفاده کنندگان از این نرم‌افزار قابل رؤیت است.
- در کنار هر کدام از نمادهای برنامه GAMS می‌توان یک متن به عنوان توضیحی از آن گنجانده که فهم مدل برای هر خواننده‌ای را امکانپذیر می‌کند. علاوه بر آن هنگام نمایش این نمادها در خروجی، توضیحات آن را نیز نمایش می‌دهد.
- تمام اطلاعاتی که برای نوشتن مدل، لازم است فقط در یک فایل گنجانده می‌شود.

هنگامی که یک معمار بنای یک ساختمان را می‌گذارد، ساختار جدید را بوسیله ایده‌ها و تکنیک‌هایی که در ساختارهای قدیمی آزموده شده‌اند را گسترش می‌دهد. این فرآیند در سایر زمینه‌ها نیز برقرار است. عناصر طراحی شده در پروژه‌های قبلی به عنوان ایده‌هایی برای تحولات جدید می‌باشند. از مراحل قبلی گسترش این نرم‌افزار مدل‌هایی به عنوان ایده برای استفاده کنندگان آن به صورت یک پایگاه اطلاعاتی در کتابخانه آن موجود است. بسیاری از مدل‌هایی که در کتابخانه نرم‌افزار موجود است از مسائل مهم کتاب‌ها درسی است که می‌توان آن را به عنوان مسائل فرموله شده در کلاس بکار برد. همچنین می‌توان از این مدل‌ها برای نمایش نکاتی کلیدی در GAMS استفاده کرد. برای شروع کار روی یکی از لینک‌های زیر کلیک کنید تا نسخه ۳۲

بیتی یا ۶۴ بیتی آن را متناسب با نوع سیستم خود دانلود نمایید و یا به آدرس www.gams.com رفته و جدید ترین نسخه آن را دانلود نمایید.

کد نویسی در GAMS:

برای فراگرفتن کد نویسی در GAMS دو مرحله پیش بینی شده است.

۱- نوشتن صحیح یک برنامه ساده در محیط GAMS.

۲- بالا بردن توانایی در نوشتن و رفع خطاهای برنامه.

در مرحله اول ابتدا ساختار کلی کد نویسی در نرم افزار GAMS آموزش داده، می شود. تمام کدهایی که در این فضا نوشته می شوند از یک چارچوب کلی پیروی می کنند. کسب توانایی های بعدی نیاز به کار بیشتر با برنامه دارد. همچنین برخی توانایی ها مربوط به دقت نظر و ظرافت در مدل سازی و استفاده از روش هایی برای نوشتن قیود به زبان GAMS می باشد و این توانایی با کار بیشتر و تجربه بدست می آید.

کافیست بتوانیم یک مدل ساده را در این نرم افزار فرموله کنیم. بعد از گذراندن این مرحله به مرور در مورد قابلیت های جانبی نرم افزار صحبت خواهیم کرد.

مثالی که برای شروع برنامه نویسی در نظر گرفته شده است. یک مدل برنامه ریزی خطی حمل و نقل است. این مثال، مثالی خوبی برای شروع برنامه نویسی و نمایش خصوصیات اساسی و جبری GAMS می باشد. مسئله برنامه ریزی خطی حمل نقل فارغ از ساینز مسئله، یک مدل ساده دارد و به سادگی با ساختارهای جبری نمایش داده می شود.

جدول زیر نمایش یک مسئله حمل و نقل است. قرار است نیاز m فروشگاه را از n انبار برآورده کنیم. در جدول زیر هزینه حمل یک واحد کالا از هر یک از انبارها به فروشگاه ها، موجودی انبار و درخواست فروشگاه ها نمایش داده شده است.

عرضه	فروشگاه m	...	فروشگاه ۲	فروشگاه ۱
------	-------------	-----	-----------	-----------

انبار ۱	c_{11}	c_{12}	...	c_{1m}	b_1
انبار ۲	c_{21}	c_{22}	...	c_{2m}	b_2
⋮	⋮	⋮		⋮	⋮
انبار n	c_{n1}	c_{n2}	...	c_{nm}	b_n
تقاضا	a_1	a_2	...	a_m	

فرموله کردن مسئله:

اندیس‌ها:

j : فروشگاه‌ها

i : انبارها

داده‌های مسئله:

a_j : تقاضای فروشگاه‌ها

b_i : موجودی انبارها

c_{ij} : هزینه حمل یک کالا از انبار i به فروشگاه j

متغیرهای تصمیم:

x_{ij} : تعداد اجناس منتقل شده از انبار i به فروشگاه j که $x_{ij} \geq 0$.

تابع هدف:

هدف کمینه کردن هزینه حمل و نقل برای این شبکه می‌باشد:

$$Z = \sum_i \sum_j c_{ij} x_{ij}$$

مجموع اجناس انتقال یافته از هر انبار نباید بیشتر از موجودی انبار باشد.

$$\sum_j x_{ij} \leq b_i \quad i = 1, 2, \dots, n$$

مجموع اجناس انتقال یافته به هر فروشگاه باید بیشتر یا برابر یا بیشتر از میزان تقاضای فروشگاه باشد.

$$\sum_i x_{ij} \geq a_j \quad j = 1, 2, \dots, m$$



تمام برنامه‌هایی که در فضای GAMS نوشته می‌شوند در این چارچوب قرار می‌گیرند. قدم اول برای شروع برنامه نویسی، نوشتن نمایش ریاضی وار مدل می‌باشد. سپس اندیس‌ها، داده‌های مسئله، متغیرها، قیود شناسایی شده تا هر کدام از آن‌ها در جای خود تعریف شوند.

- از مجموعه‌ها (Set) برای تعریف اندیس‌ها استفاده می‌کنیم.
- از اسکالر (Scalar) برای داده‌هایی از مسئله استفاده می‌کنیم که اصطلاحاً گفته می‌شود داده بدون بعد باشد. برای مثال برای معرفی $a=1$ به برنامه، باید در قسمت Scalar این اتفاق رخ دهد.
- از پارامتر (Parameter) برای معرفی داده‌های جدولی با یک بعد استفاده می‌شود. همچنین ممکن است بخواهیم جواب مسئله را در پارامترهایی ذخیره کنیم تا در مراحل بعد استفاده شود. در اینصورت نیز برای معرفی آن پارامتر از گزاره Parameter استفاده می‌شود (نکته: تمامی اسکالرها را در قالب پارامتر نیز می‌توان استفاده کرد).
- اگر داده‌های مسئله به صورت جداولی با بیش از یک بعد باشند، برای معرفی آن از گزاره Table استفاده می‌شود.
- متغیرهایی که در مسئله به کار می‌روند بر حسب نوع آن‌ها در قالب گزاره Positive Variable, Variable, Binary Variable, Integer Variable و Negative Variable تعریف می‌شوند. لازم به ذکر است که متغیر هدف حتماً باید به صورت متغیر آزاد (Free Variable یا Variable) تعریف شود.
- معرفی قیود به برنامه در دو مرحله انجام می‌شود. ابتدا در قالب گزاره Equations فقط تعریف می‌شوند. یعنی هر قیدی که در مسئله استفاده می‌شود باید با یک اسم مشخص شود. بعد از معرفی قیود، آن‌ها را به کمک علامت ریاضی برنامه تعریف می‌کنیم.

برای مسئله‌ای که قصد حل کردن آن را داریم باید یک اسم اختصاص دهیم و قیودی که در این مسئله دخیل هستند را مشخص کنیم. این کار بوسیله گزاره Model صورت می‌گیرد.

برای فراخوانی مسئله جهت حل آن از گزاره Solve استفاده می‌کنیم. در این گزاره اسم مدلی که قرار است حل شود، نحوه حل (کمینه یا بیشینه)، نوع برنامه (خطی، غیرخطی و غیره) و متغیر هدف مشخص می‌شود.

نکات زیر را در کد نویسی در فضای GAMS به خاطر داشته باشیم.

۱- هیچ محدودیتی در استفاده از حروف بزرگ بجای حروف کوچک و بالعکس نیست. برای مثال کلمات New-York، new-York و NEW-YORK نیست.

۲- هر گزاره‌ای باید با یک علامت ";" پایان یابد. برای جدا کردن موارد مشابه از " " استفاده می‌کنیم.

بعد از تعریف اسامی اتخاذ شده برای اسکالرها، پارامترها، متغیرها، قیود و ... می‌توان با ایجا یک کارکتر فاصله (Space) توضیحی را در مورد آن و برای فهم بهتر مدل توسط اشخاص دیگر بدنبال آن آورد که در این صورت این متن به رنگ آبی ظاهر خواهد شد. و GAMS به مانند یک توضیح جانبی در مورد آن پارامتر، متغیر و یا ... با آن برخورد خواهد کرد.



مثال:

داده‌های زیر مربوط به مسافت بین انبارها و فروشگاه‌ها و هزینه جابجایی یک واحد کالا می‌باشد.

فروشگاه‌ها انبارها	فاصله انبارها از فروشگاه (هزار مایل)			عرضه (موجودی)
	New York	Chicago	Topeka	
Seattle	۲,۵	۱,۷	۱,۸	۳۵۰
San Diego	۲,۵	۱,۸	۱,۴	۶۰۰
تقاضا	۳۲۵	۳۰۰	۲۷۵	

هزینه انتقال هر واحد کالا برای هر هزار مایل ۹۰ دلار می‌باشد.

۱- اندیس‌ها:

i : Seattle و San Diego

j : Topeka و Chicago، New York

همانطور که قبلاً نیز گفته شد، اندیس‌ها باید در قالب گزاره Set به برنامه معرفی شود. ابتدا عبارت Set سپس اسم اندیس (i) بعد از یک فاصله توضیحی در مورد آن (اختیاری) و معرفی اعضای آن بین دو عبارت "//".

Sets

i canning plants / seattle, san-diego/

j markets / new-york, chicago, topeka/

;

و یا

Sets i canning plants / seattle, san-diego/ , j markets / new-york, chicago, topeka/

;

و یا (توضیحات اختیاری است):

Sets i / seattle, san-diego/

j / new-york, chicago, Topeka/

;

همانطور که قبلاً نیز گفته شد برای تعریف چند مجموعه، آنها را باید در سطور جداگانه نوشت و یا با یک علامت ویرگول آنها را از هم جدا کرد.

۲- داده‌های مسئله:

داده‌هایی که در مسئله استفاده می‌شوند باید در قالب پارامتر، اسکالر یا جدول تعریف شوند. در اینج ۴ ورودی داریم. عرضه و تقاضا (داده‌های تک بعدی)، هزینه حمل کالا در هر هزار مایل (داده اسکالر) و هزینه حمل کالا از انبارها به فروشگاه‌ها که از ضرب هزینه در فاصله دست می‌آید.

۱-۲ اسکالر

برای معرفی اسکالر (داده بدون بعد) ابتدا کلمه Scalar سپس اسم اسکالر و توضیحی در مورد آن و در انتها مقدار آن بین دو عبارت "//" نوشته می‌شود:

Scalar f freight in dollars per case per thousand miles /۹۰/;

و یا (توضیحات اختیاری است):

Scalar f /۹۰/;

۲-۲ پارامترها:

پارامترها به دو گونه مشخص می‌شوند. به صورت مستقیم و به صورت فرمولی. برای معرفی پارامترها به صورت مستقیم ابتدا اسم پارامتر مشخص شود، سپس اندازه آن را داخل پرانتز آورده (اجباری)، با ایجاد یک فاصله توضیحی در مورد آن می‌دهیم (اختیاری) و برای هر کدام از اندیس‌ها مقدار آن را داخل "//" می‌نویسیم.

Parameters

a(i) capacity of plant i in cases

/ seattle ۳۵۰

san-diego ۶۰۰ /

b(j) demand at market j in cases

/ new-york ۳۲۵

chicago ۳۰۰

topeka ۲۷۵ /

;

ویا

Parameters a(i) capacity of plant i in cases / seattle ۳۵۰, san-diego ۶۰۰ /

b(j) demand at market j in cases / new-york ۳۲۵, chicago ۳۰۰, topeka ۲۷۵ /

;

ویا (توضیحات اختیاری است):

Parameters a(i) /seattle ۳۵۰, san-diego ۶۰۰ /, b(j) /new-york ۳۲۵, chicago ۳۰۰, topeka ۲۷۵ /;

اندیس داخل پرانتز نشان می‌دهد که پارامتر a باید برای هر کدام از اندیس‌های مجموعه i یک مقدار بگیرد. بعد از معرفی نام پارامتر برای هر کدام از اندیس‌ها مقادیر مربوط به آن را روبروی آن و در همان سطر می‌نویسیم. بحث مشابهی در مورد اندیس متناظر پارامتر b نیز وجود دارد.

برای مشخص کردن پارامتر به صورت فرمولی ابتدا در قالب پارامتر تنها نام آن مشخص می‌شود و بعد رابطه ریاضی مربوط به آن مشخص می‌شود. در اینجا هزینه حمل کالا از انبارها به فروشگاه‌ها از حاصلضرب هزینه حمل هر کالا (در هزار مایل) در فاصله بین فروشگاه و انبار بدست می‌آید. اما از آنجا که فاصله‌ها هنوز به مدل معرفی نشده‌است، لذا معرفی آن را به تأخیر می‌اندازیم.

۲-۲ جداول:

از جدول (Table) برای معرفی کردن داده‌هایی با بیش از یک بعد استفاده می‌شود. برای اینکار ابتدا کلمه Table بعد اسم جدول و اندیس‌های مربوط به آن (اندس‌های سطور و ستون‌ها آن)، سپس توضیح مختصری در مورد آن (اختیای) و در انتها با ساخت یک جدول مقادیر آن را وارد می‌کنیم. می‌خواهیم فاصله بین انبارها و فروشگاه‌ها را وارد کنیم. یک جدول ۲ در ۳ که سطرهای آن مربوط به انبارها و ستون‌های آن مربوط به فروشگاه‌ها است.

Table d(i,j) distance in thousands of miles

	new-york	chicago	topeka
seattle	۲,۵	۱,۷	۱,۸
san-diego	۲,۵	۱,۸	۱,۴

ویا (توضیحات اختیاری است):

Table d(i,j)

	new-york	chicago	topeka
seattle	۲,۵	۱,۷	۱,۸
san-diego	۲,۵	۱,۸	۱,۴

دقت کنید که در نوشتن جدول عناوین سطری باید از مجموعه اول و عناوین ستون جدول باید از مجموعه دوم انتخاب شود. در غیر اینصورت GAMS خطای ۱۷۰ (Domain violation for element) را به شما خواهد داد. که نشان می‌دهد شما در نوشتن عناوین سطری یا ستونی از مجموعه متناظر استفاده نکرده‌اید. حال که جدول تعریف شد، پارامتر هزینه که به صورت فرمولی تعریف می‌شود را توضیح می‌دهیم. همانطور که گفتیم، ابتدا تنها نام آن را در گزاره پارامتر تعریف می‌کنیم بعداً (قبل از دستور اجرای برنامه یعنی Solve) آن را تعریف می‌کنیم:

Parameter c(i,j) transport cost in thousands of dollars per case;

$$c(i,j) = f * d(i,j) / 1000;$$

۳- متغیرها:

$$x_{ij} : \text{تعداد اجناس منتقل شده از انبار } i \text{ به فروشگاه } j \text{ که } x_{ij} \geq 0$$

Positive Variable

$x(i,j)$ shipment quantities in cases
;

ویا (توضیحات اختیاری است):

Positive Variable

$x(i,j)$
;

متغیر تابع هدف که هزینه کل را کمینه می کند، باید به صورت متغیر آزاد تعریف شود.

Variables

Z total transportation costs in thousands of dollars
;

ویا (توضیحات اختیاری است):

Variables

Z
;

۴- قیود

قیودی که برای مسئله تعریف می شوند ابتدا باید نامگذاری شوند. در اینجا دو قید برای انبارها و سه قید برای فروشگاهها داریم. نیازی نیست که پنج قید معرفی شوند. دو قید ابتدایی تنها فرقیان در دادههای یکبار رفته در آن است. برای قید اول دادههای مربوط به انبار اول و برای قید دوم دادههای مربوط به انبار دوم استفاده می شود. به عبارت دیگر جنس این قیود، یکی است. یعنی یک قید از سائز انبارها و قید دیگر از سائز فروشگاهها.

پس:

Equations

cost define objective function

supply(i) observe supply limit at plant i

demand(j) satisfy demand at market j ;

ویا (توضیحات اختیاری است):

Equations

Cost, supply(i), demand(j) ;

در مرحله بعد این قیدها را به فرم ریاضیاتی می نویسم. بدین صورت که ابتدا نام قید، سپس سائز آن، در مرحله بعد علامت " .. " و سپس نمایش آن به زبان ریاضی. قیود کوچکتر یا مساوی را با =L=، قیود مساوی را با =E= و قیود بزرگتر یا مساوی را با =G= نمایش می دهیم:

Cost .. $z = e = \text{sum}((i,j), c(i,j)*x(i,j)) ;$

supply(i) .. $\text{sum}(j, x(i,j)) = l = a(i) ;$

demand(j) .. $\text{sum}(i, x(i,j)) = g = b(j) ;$

منظور از supply(i) این است که این قید، به تعداد هر کدام از اعضای مجموعه i نوشته می شود. یعنی مدل با انتخاب هر کدام از اعضای مجموعه i متغیرها و پارامترهای همان عضو را در قد در نظر می گیرد. تفسیر مشابهی برای demand(j) نیز وجود دارد.

۵- انتخاب نام برای مدل و مشخص کردن قیدهایی که در آن نقش دارند.

بعد از کلمه مدل نامی برای مسئله انتخاب می شود و قیود معرفی شده در بالا که در مسئله نقش دارند داخل علامت "/" قرار می گیرد.

Model transport /all/ ;

اسم مدل Transport و منظور از all یعنی تمامی قیود معرفی شده در مدل نقش دارند. همچنین می توان نوشت:

Model Transport /Cost, Demand, Supply/ ;

۶- حل مسئله:

بعد از کلمه Solve به ترتیب نام مدل، کلمه Using، نوع مدل (Lp، Nlp یا ...)، نوع مسئله (کمینه سازی یا بیشینه سازی) و در نهایت متغیر هدف می آید.

Solve Transport Using Lp Minimizing z;

۷- نمایش نتایج:

برای نمایش هر کدام از عناصر دلخواه می توان از Display استفاده کرد. برای نمایش پارامترها تنها از اسم آنها ولی برای نمایش متغیرها از اسم متغیر و پسوند l یا m استفاده می شود که اولی مربوط به مقدار بهینه متغیر در مسئله پرایمال و دومی، مقدار بهینه متغیر در مسئله ثانویه است.

Display Z.l, X.l, X.m;

پیدا سازی