

ການນຳໃຊ້ MATLAB ແກ້ບັນຫາການຂົນສົ່ງ

ຜູ້ ສຸກສິມຫວັງ¹ ແລະ ແສງດາລາ ບົວລາຄຳ²

ພາກວິຊາຄະນິດສາດ ແລະ ສະຖິຕິ, ຄະນະວິທະຍາສາດທຳມະຊາດ, ມະຫາວິທະຍາໄລແຫ່ງຊາດ

ບົດຄັດຫຍໍ້

¹ ຜູ້ຕິດຕໍ່ພົວພັນ: ຜູ້ ສຸກສິມຫວັງ,
ພາກວິຊາຄະນິດສາດ ແລະ ສະຖິຕິ,
ຄະນະວິທະຍາສາດທຳມະຊາດ,
ມະຫາວິທະຍາໄລແຫ່ງຊາດ
ເບີໂທ: +856 20 5299 4619,
ອີເມວ: phoui20@gmail.com

² ພາກວິຊາຄະນິດສາດທຳມະຊາດ,
ຄະນະສຶກສາສາດ,
ມະຫາວິທະຍາໄລສຸພານຸວົງ

ຂໍ້ມູນບົດຄວາມ:

ການສົ່ງບົດຄວາມ: 12 ກໍລະກົດ 2021
ປັບປຸງສຳເລັດ: 26 ສິງຫາ 2021
ການຕອບຮັບ: 17 ກັນຍາ 2021

ຢູ່ໃນບັນຫາລິເນແອໂປຣແກຣມມິງ, ແບບຈຳລອງທາງຄະນິດສາດຂອງບັນຫາການຂົນສົ່ງເປັນພື້ນຖານຂອງການແກ້ໄຂຫຼາຍບັນຫາໃນຊີວິດຈິງທີ່ກ່ຽວຂ້ອງກັບບັນຫາອັອບຕິໄມເຊຊັນ. ມັນເປັນສິ່ງທີ່ຈະແຈ້ງແລ້ວວ່າຜ່ານມາໄດ້ມີນັກຄະນິດສາດຫຼາຍທ່ານພະຍາຍາມຊອກຫາວິທີແກ້ໄຂບັນຫາການຂົນສົ່ງເພື່ອໃຫ້ໄດ້ຄຳຕອບທີ່ດີທີ່ສຸດ. ວິທີເຫຼົ່ານັ້ນ ລວມມີວິທີ Northwest Corner Rule, Least Cost Method, Vogel's Approximation Method, and Modi method ແຕ່ວ່າໃນປັດຈຸບັນນັກຄົ້ນຄວ້າສ່ວນຫຼາຍຈະໃຊ້ ການຄິດໄລ່ດ້ວຍມື ຫລື ໃຊ້ໂປຣແກຣມສຳເລັດຮູບເຊັ່ນ: LINDO, QM ຫລື EXCEL ເຂົ້າຊ່ວຍໃນການຊອກຫາໃຈຜົນທີ່ເໝາະສົມ ແຕ່ວ່າການນຳໃຊ້ໂປຣແກຣມສຳເລັດຮູບເຫຼົ່ານັ້ນເຂົ້າແກ້ບັນຫາພວກເຮົາຈະໄດ້ແຕ່ຄຳຕອບ, ຈະບໍ່ເຂົ້າໃຈເລິກເຊິ່ງເຖິງຂະບວນການຂອງການຄິດໄລ່. ໃນທາງກົງກັນຂ້າມຖ້າພວກເຮົາໃຊ້ການຂຽນໂຄດໃນ MATLAB ເຂົ້າແກ້ບັນຫາດັ່ງກ່າວ ຈະເຮັດໃຫ້ພວກເຮົາເຂົ້າໃຈເລິກເຊິ່ງໃນຂັ້ນຕອນຕ່າງໆຂອງການຊອກຫາໃຈຜົນຂອງບັນຫາ ແລະ ມັນຍັງຊ່ວຍແກ້ບັນຫາໃນກໍລະນີທີ່ມາຕຣິສຂອງຄ່າໃຊ້ຈ່າຍມີຂະໜາດໃຫຍ່ຢ່າງຍືນຍົງ. ດັ່ງນັ້ນ, ໃນບົດວິຈານີ້ຜູ້ຂຽນຈະນຳໃຊ້ການຂຽນໂຄດໃນ MATLAB ເຂົ້າແກ້ໄຂບັນຫາການຂົນສົ່ງ ໂດຍມີເປົ້າໝາຍແກ້ບັນຫາຂົນສົ່ງຕາມວິທີ Northwest Corner Rule, Least Cost Method, Vogel's Approximation Method, and Modi method

ຄຳສັບສຳຄັນ: ແບບຈຳລອງທາງຄະນິດສາດ, ບັນຫາການຂົນສົ່ງ, MATLAB code, North West Corner Rule, Least Cost Method, and Vogel's Approximation Method

Solves Transportation Problems Using MATLAB

Phoui Souksomvang and Sengdala Boualakhom

Department of Mathematics and Statistics, Faculty of Natural Sciences, National University of Lao, Lao PDR

ABSTRACT

¹**Correspondence:** Phoui Souksomvang, Department of Mathematics and Statistics, Faculty of Natural Sciences, National University of Lao, Tel: +856 20 5299 4619, E-mail: phoui20@gmail.com

² Department of Natural Science Teachers, Faculty of Education, Souphanouvong University

Article Info:

Submitted: Jul 12, 2021

Revised: Aug 26, 2021

Accepted: Sep 17, 2021

In the linear problem of programming, the mathematical model of the transport problem is the basis for solving many real-life problems related to optimization problems. It is clear that in the past many mathematicians have tried to find solutions to the transportation problem to get the best answer. Those methods include the Northwest Corner Rule, the Least Cost Method, Vogel's Approximation Method, and the Modi method, but now most researchers use manual computation or a built-in program such as LINDO, QM or EXCEL to help you figure out the solution. On the other hand, if we use coding in MATLAB to solve the problem, it will give us a deeper understanding of the various stages of the search for the outcome of the problem, and it also helps to solve the problem in case the matrix of costs is much larger. Therefore, in this study, the author will use coding in MATLAB to solve transportation problems with the goal of solving transportation problems according to the Northwest Corner Rule, Least Cost Method, Vogel's Approximation Method, and Modi method.

Keywords: *Mathematical Modeling, Transportation Issues, MATLAB code, North West Corner Rule, Least Cost Method, and Vogel's Approximation Method.*

1. ພາກສະເໜີ

ບັນຫາການຂົນສົ່ງໃນຂົງເຂດຂອງການວິໄຈດຳເນີນງານແມ່ນໜຶ່ງໃນບັນຫາລິເນແອໂປຣແກຣມມິງ ທີ່ມີການພັດທະນາແຕ່ດົນນານແລ້ວໂດຍທ່ານ Hitchcock (Anderson & T.A, 1982). ຈາກທຳມະຊາດຂອງບັນຫາ ແລະ ອີງໃສ່ ຊົມເປລັກສ ອາລກໍຣິດ, ມີຫລາຍວິທີທີ່ມີປະສິດທິຜົນໃນການຊອກຫາໃຈຜົນໄດ້ຮັບການພັດທະນາຂຶ້ນ (Alexande, 1998) ແລະ (Dantizg, 1963) ຄ້າຍຄືກັນກັບ ຊົມເປລັກສ ອາລກໍຣິດ, ຜົນຮັບພື້ນຖານຂັ້ນຕົ້ນກໍມີຄວາມຕ້ອງການຄືກັນໃນບັນຫາຂອງການຂົນສົ່ງ ໃນບົດວິໄຈນີ້ວິທີການຕ່າງໆເຊັ່ນ: Northwest Corner

Rule, Least Cost ແລະ the Vogel Approximation ຈະຖືກນຳໃຊ້.

ໃນທິດສະດີ ແລະ ພາກປະຕິບັດຕົວຈິງ, ບັນຫາການຂົນສົ່ງທັງໝົດສາມາດສ້າງເປັນແບບຈຳລອງທາງຄະນິດສາດໃນຮູບແບບຂອງຮູບຮ່າງມາດຕະຖານຂອງບັນຫາລິເນແອໂປຣແກຣມມິງທີ່ສາມາດນຳໃຊ້ວິທີຊົມເປລັກສຊອກຫາໃຈຜົນໄດ້. ສະນັ້ນ, ຈຶ່ງເວົ້າໄດ້ວ່າບັນຫາການຂົນສົ່ງແມ່ນກໍລະນີພິເສດຂອງບັນຫາລິເນແອໂປຣແກຣມມິງທີ່ສາມາດຊອກຫາໃຈຜົນທີ່ດີທີ່ສຸດໄດ້ຈາກວິທີຂອງ Steeping Stone Algorithm ຫຼື Modified Distribution Method (MODI).

ວິທີ Steepest Descent ແມ່ນຖືກພັດທະນາຂຶ້ນທຳອິດ Charnes et. (Anderson & T.A, 1982). ໂດຍມີຄວາມຕັ້ງໃຈໃຫ້ເປັນທາງເລືອກໜຶ່ງໃນການຊອກຫາໃຈຜົນທີ່ດີທີ່ສຸດຂອງບັນຫາການຂົນສົ່ງ. ແຕ່ວ່າ LINDO (Linear Interactive and Discrete Optimization) ເປັນໂປຼແກຼມສຳເລັດຮູບທີ່ສາມາດຊອກຫາໃຈຜົນຂອງບັນຫາການຂົນສົ່ງຢ່າງມີປະສິດທິພາບໃນກໍລະນີທີ່ບັນຫານັ້ນຢູ່ໃນຮູບຮ່າງມາດຕະຖານຂອງບັນຫາລິເນແອໂປຼແກຼມມິງ.

ໃນປັດຈຸບັນນີ້ MATLAB ໄດ້ຖືກນຳມາໃຊ້ຫຼາຍໃນຂົງເຂດຂອງວິທະຍາສາດ ໂດຍສະເພາະແມ່ນ ຄະນິດສາດ ແລະ ຟີຊິກສາດ ແຕ່ວ່າການຂຽນ MATLAB ໂຄດແກ້ບັນຫາການຂົນສົ່ງນີ້ຍັງມີໜ້ອຍ ທັງໆທີ່ MATLAB ໂຄດສາມາດເຮັດໃຫ້ການແກ້ບັນຫານັ້ນງ່າຍຂຶ້ນ ໂດຍສະເພາະກໍລະນີທີ່ມາດຕະຖານຂອງຄ່າໃຊ້ຈ່າຍມີຂະໜາດໃຫຍ່. ດັ່ງນັ້ນ, ພວກເຮົາຈຶ່ງມີຄວາມສົນໃຈທີ່ຈະຂຽນ MATLAB ໂຄດ ເພື່ອຊອກຫາໃຈຜົນທີ່ດີທີ່ສຸດຂອງບັນຫາການຂົນສົ່ງ.

ຈຸດປະສົງການຄົ້ນຄວ້າຄັ້ງນີ້ ແມ່ນນຳໃຊ້ການຂຽນໂຄດໃນ MATLAB ເຂົ້າແກ້ໄຂບັນຫາການຂົນສົ່ງ ໂດຍມີເປົ້າໝາຍແກ້ບັນຫາຂົນສົ່ງຕາມວິທີ Northwest Corner Rule, Least Cost Method, Vogel's Approximation method, and Modi method

2. ອຸປະກອນ ແລະ ວິທີການ

2.1 ການອອກແບບຂອງບັນຫາການຂົນສົ່ງ

ພິຈາລະນາ m ຈຸດຕົ້ນທາງ (ຈຳນວນໂຮງງານການຜະລິດ) S_i ($i = 1, \dots, m$) ແລະ n ຈຸດປາຍທາງ (ຈຳນວນຄັ້ງສົນຄ້າ ຫຼື ຈຳນວນລູກຄ້າ) D_j ($j = 1, \dots, n$), ໃຫ້ a_i ແມ່ນຈຳນວນສິນຄ້າທີ່ມີຢູ່ຈຸດຕົ້ນທາງ S_i ໄປຫາປາຍທາງ D_j ເພື່ອໃຫ້ຕອບສະໜອງກັບຄວາມຕ້ອງການ b_j ຂອງຈຳນວນສິນຄ້າທີ່ມີຢູ່ຈຸດປາຍທາງ, C_{ij} ແມ່ນຄ່າໃຊ້ຈ່າຍໃນການຂົນສົ່ງສິນຄ້າໜຶ່ງໜ່ວຍຈາກຈຸດຕົ້ນທາງທີ i ໄປຍັງຈຸດປາຍທາງທີ j , x_{ij} ແມ່ນຈຳນວນສິນຄ້າທີ່ສົ່ງຈາກຈຸດຕົ້ນທາງທີ i ໄປຍັງຈຸດປາຍທາງທີ j , Z ແມ່ນຄ່າໃຊ້ຈ່າຍລວມ.

ບັນຫາທີ່ເວົ້າມາເທິງນີ້ສາມາດຂຽນເປັນແບບຈຳລອງທາງຄະນິດສາດທີ່ເວົ້າເຖິງການຂົນສົ່ງໄດ້ດັ່ງນີ້:

$$\text{Minimize}(Z) = \sum_{i=1}^m \sum_{j=1}^n C_{ij} x_{ij}$$

Subject to constraint

$$\sum_{j=1}^n x_{ij} = a_i, \quad i = 1, 2, \dots, m \text{ ແມ່ນຂີດຈຳກັດຂອງ}$$

ສິນຄ້າທີ່ມີຢູ່ຈຸດຕົ້ນທາງ

$$\sum_{i=1}^m x_{ij} = b_j, \quad j = 1, 2, \dots, n \text{ ແມ່ນຂີດຈຳກັດຂອງ}$$

ສິນຄ້າທີ່ມີຢູ່ຈຸດປາຍທາງ

$$x_{ij} \geq 0, \quad i = 1, 2, \dots, m; \quad j = 1, 2, \dots, n$$

$$\sum_{i=1}^m a_i = \sum_{j=1}^n b_j \text{ ແມ່ນເງື່ອນໄຂສຳລັບຄວາມເທົ່າກັນ}$$

ລະຫວ່າງຄວາມຕ້ອງການ ແລະ ການຕອບສະໜອງ ຊຶ່ງເປັນເງື່ອນໄຂທີ່ເຮັດໃຫ້ບົດເລກປະກົດຜົນຮັບຟື້ນຖານຂຶ້ນຕື່ມ.

2.2 ການຊອກຫາໃຈຜົນ

A. ການຕັ້ງຜົນຮັບຟື້ນຖານຂຶ້ນຕື່ມດ້ວຍວິທີຂອງ Northwest Corner

ການຕັ້ງຜົນຮັບເບື້ອງຕົ້ນດ້ວຍວິທີຂອງ Northwest Corner ເປັນວິທີທີ່ງ່າຍ ໂດຍຈະພິຈາລະນາໃນດ້ານຈຳນວນສິນຄ້າເທົ່ານັ້ນ ໂດຍບໍ່ພິຈາລະນາດ້ານຄ່າໃຊ້ຈ່າຍໃນການຂົນສົ່ງເລີຍ, ເຊິ່ງຂັ້ນຕອນໃນການຕັ້ງຜົນຮັບເບື້ອງຕົ້ນຈະມີຂັ້ນຕອນດັ່ງນີ້:

1. ເລີ່ມການຄິດໄລ່ຈາກຫ້ອງທີໜຶ່ງເບື້ອງຊ້າຍມື ຄື: ຫ້ອງທີ (1,1)
2. ກຳນົດຄ່າ $x_{11} = \min(a_1, b_1)$
3. ຫັກຄ່າ x_{ij} ທີ່ຈັດສັນອອກຈາກຄ່າ a_i ແລະ b_j
4. ຖ້າວ່າຄ່າ a_i ເຫຼືອໃຫ້ເລື່ອນໄປຈັດຫ້ອງຫວ່າງທາງຂວາມື, ຖ້າວ່າ b_j ເຫຼືອໃຫ້ເລື່ອນໄປຈັດໃສ່ຫ້ອງຫວ່າງທາງລຸ່ມ
5. ກຳນົດຄ່າຂອງ $x_{ij} = \min(a_i, b_j)$
6. ກັບໄປຂັ້ນຕອນທີ 3

3. ຜົນໄດ້ຮັບ

ໃນການຄົ້ນຄວ້າຄັ້ງນີ້ແມ່ນນຳໃຊ້ການຂຽນໂຄດໃນ MATLAB ເຂົ້າແກ້ໄຂບັນຫາການຂົນສົ່ງໂດຍມີເປົ້າໝາຍ ແກ້ບັນຫາຂົນສົ່ງຕາມວິທີ Northwest Corner Rule, Least Cost Method, Vogel's Approximation Method, and Modi method ໂດຍແບບຈຳລອງທາງຄະ

ນິດສາດຕົວຢ່າງດັ່ງຕໍ່ໄປນີ້:

ຕົວຢ່າງ: ຈົ່ງຊອກຫາຜົນຮັບພື້ນຖານຂັ້ນຕົ້ນຂອງບັນຫາ ການຂົນສົ່ງລຸ່ມນີ້ໂດຍໃຊ້ວິທີຂອງ Northwest Corner (ໜ່ວຍ 10,000 ກີບ)

ໂຮງງານ	ສາຍເກັບສິນຄ້າທີ				
	1	2	3	4	a_i
1	1	2	2	5	500
2	3	5	2	4	300
3	4	1	3	3	600
b_j	400	200	500	300	1,400

ຈົ່ງຕັ້ງຜົນຮັບເບື້ອງຕົ້ນດ້ວຍວິທີຂອງ Northwest Corner

Matlab M-File 1

```
a=[1 2 2 5;3 5 2 4;4 1 3 3];
supply=[500 300 600];
demand=[400 200 500 300];
[m n]=size(a)
k=zeros(m,n);
```

```
sum=0;
% main loop
i=1;j=1;
while i<=m&&j<=n
    min1=min(demand(j),supply(i));
    sum=sum+(a(i,j).*min1);
    demand(j)=demand(j)-min1;
    supply(i)=supply(i)-min1;
    if demand(j)==0
        j=j+1;
    end
    if supply(i)==0
        i=i+1;
    end
end
disp(['the cost of the North West Method is ',num2str(sum)])
```

ຫລັງຈາກ run ໂຄດ Matlab M-File 1 ຈະໄດ້ຄຳຕອບ $3,300 \times 1000 = 33,000,000$ ກີບ

B. ການຕັ້ງຜົນຮັບພື້ນຖານຂັ້ນຕົ້ນດ້ວຍວິທີຂອງ Least Cost method

ການຕັ້ງຜົນຮັບເບື້ອງຕົ້ນດ້ວຍວິທີ **Least Cost Method** ເປັນວິທີ ທີ່ນຳຄ່າໃຊ້ຈ່າຍໃນການຂົນສົ່ງທັງໝົດ ມາພິຈາລະນາ, ໂດຍເລືອກຫ້ອງທີ່ມີຄ່າຂົນສົ່ງຕໍ່າສຸດໃນຕາ ຕະລາງ, ເຊິ່ງມີຂັ້ນຕອນດັ່ງນີ້:

1. ເລືອກຫ້ອງຂອງ c_{ij} ທີ່ມີຄ່າຕໍ່າສຸດ.
2. ກຳນົດຄ່າ $x_{ij} = \min(a_i, b_j)$ ໃສ່ໃນທີ່ເລືອກໄວ້.

3. ຖ້າວ່າ $a_i < b_j$ ສະແດງວ່າຈຳນວນສິນຄ້າທີ່ຈັດໃສ່ ຫ້ອງທີ່ເລືອກໄວ້ເທົ່າກັບ a_i ແລ້ວຕັດ ແຖວນອນທີ່ i ອອກ ແລ້ວກັບໄປຂັ້ນຕອນທີ 1 ໃໝ່, ໂດຍພິຈາລະ ນາຈາກຕາຕະລາງທີ່ເຫຼືອ, ເຊິ່ງຈຳນວນສິນຄ້າທີ່ເຫຼືອໃນ ແຖວຕັ້ງທີ j ເທົ່າກັບ $b_j - a_i$. ຖ້າວ່າ $a_i > b_j$ ສະ ແດງວ່າຈຳນວນສິນຄ້າທີ່ຈັດໃສ່ຫ້ອງທີ່ເລືອກໄວ້ເທົ່າກັບ b_j ແລ້ວຕັດແຖວ ຕັ້ງທີ j ອອກ, ເຊິ່ງຈຳນວນສິນຄ້າ ທີ່ເຫຼືອໃນແຖວນອນທີ່ i ເທົ່າກັບ $a_i - b_j$ ແລ້ວກັບ

ໄປຂັ້ນຕອນທີ 1 ໃໝ່, ໂດຍພິຈາລະນາຈາກຕາຕະລາງທີ່ເຫຼືອ.

ເຫຼືອຈະເທົ່າກັບຈຳນວນທີ່ເຫຼືອຢູ່ໃນແຖວນອນ ຫຼື ແຖວຕັ້ງຂອງຫ້ອງນັ້ນ.

4. ເຮັດຕາມຂັ້ນຕອນທີ 1 ເຖິງ 3 ໄປເລື້ອຍໆ ໃຫ້ເຫຼືອພຽງແຖວນອນ ຫຼື ແຖວຕັ້ງໜຶ່ງແຖວ. ຈຳນວນສິນຄ້າທີ່

ຕົວຢ່າງ: ຈົ່ງຊອກຫາຜົນຮັບພື້ນຖານຂັ້ນຕົ້ນຂອງບັນຫາການຂົນສົ່ງລຸ່ມນີ້ໂດຍໃຊ້ວິທີຂອງ **Least Cost Method** (ໜ່ວຍ 10,000 ກີບ)

ໂຮງງານ	ລູກຄ້າ					Supply
	1	2	3	4		
1	1	2	3	4	8	6
2	4	3	2	0	0	8
3	0	2	2		1	10
Demand	4	6	8	6	24	

ຈົ່ງຕັ້ງຜົນຮັບເບື້ອງຕົ້ນດ້ວຍວິທີຂອງ Least Cost

```
if (X==demand(j(1)))
    A(:,j(1))=[];
    demand(j(1))=[];
```

Method.

Matlab M-File 2

```
close all;
clear all;
clc
A=[1 2 3 4;4 3 2 0; 0 2 2 1];
supply=[6 8 10];
demand=[4 6 8 6];
y=0;
while (~isempty(supply)
&&~isempty(demand))
    M=min(A);
    N=min(M);
    i=0;
    j=0;
    [i,j]=find(A==N);
    X=min(supply(i(1)),demand(j(1)))
    ;
    y=y+X+N;
    if (X==supply(i(1)))
        A(i(1),:)=[];
        supply(i(1))=[];
    demand(j(1))=demand(j(1))-X;
end
```

```
supply(i(1))=supply(i(1))-X;
end
end
disp('The cost is of the given
input matrix is:')
disp(y)
```

ຫລັງຈາກ run ໂຄດ Matlab M-File 2 ຈະໄດ້ຄຳຕອບ $3,300 \times 1000 = 330,000$ ກີບ

C. ການຕັ້ງຜົນຮັບພື້ນຖານຂັ້ນຕົ້ນດ້ວຍວິທີຂອງ Vogel Approximation Method (VAM)

ລະບຽບວິທີປະມານຄ່າໂອເກລ (Vogel's) ເປັນການຕັ້ງຜົນຮັບເບື້ອງຕົ້ນ ທີ່ມີວິທີການຄຳນວນທີ່ຫຍຸ້ງຍາກກ່ວາວິທີອື່ນໆ ເນື່ອງຈາກເປັນວິທີທີ່ນຳເອົາຄ່າໃຊ້ຈ່າຍໃນການຂົນສົ່ງຂອງທຸກໆ ໂຮງງານເຂົ້າມາພິຈາລະນາຮ່ວມກັນມີຂັ້ນຕອນຄຳນວນດັ່ງນີ້:

1. ຄຳນວນຜົນລະຫວ່າງຄ່າໃຊ້ຈ່າຍ 2 ຕົວທີ່ຕ່ຳທີ່ສຸດຂອງແຕ່ລະແຖວນອນ ແລະ ແຖວຕັ້ງ
2. ເລືອກແຖວນອນ ຫຼື ແຖວຕັ້ງທີ່ມີຄ່າຜົນລະຫວ່າງທີ່ສຸດ

3. ເລືອກຫ້ອງທີ່ມີຄ່າໃຊ້ຈ່າຍຕໍ່າທີ່ສຸດໃນແຖວທີ່ເລືອກໄວ້
4. ກຳນົດຄ່າ $x_{ij} = \min(a_i, b_j)$
5. ຕັດແຖວນອນ ຫຼື ແຖວຕັ້ງທີ່ຈັດສັນຄ່າ a_i ຫຼື b_j ໝົດ ແລ້ວອອກໄປ

6. ກັບໄປ 1

ຕົວຢ່າງ: ຈົ່ງຊອກຫາຜົນຮັບພື້ນຖານຂັ້ນຕົ້ນຂອງບັນຫາການຂົນສົ່ງລຸ່ມນີ້ໂດຍໃຊ້ວິທີຂອງ Vogel Approximation method (VAM) (ໜ່ວຍ 10,000 ກີບ)

ຈົ່ງຕັ້ງຜົນຮັບເບື້ອງຕົ້ນດ້ວຍວິທີຂອງ VAM

```
function [ z,x ] = VAM(s,d,c,m,n)
x=zeros(m,n);
numbasic=n+m-1;
for j=1:n
    for i=1:m
        c1(i,j)=c(i,j);
    end
end
for i=1:m
    s1(i)=s(i);
end
for j=1:n
    d1(j)=d(j);
end
%% Vogel's approximation
method
iteration=0;
for k=1:n+m-1% the mistaken code
to be " for k= 1:n+m"
    iteration=iteration+1;
%% Row Difference
minrow1=zeros(m,1);
minrow2=zeros(m,1);
jmin=zeros(1,m);
for i=1:m
    min1=inf;
    for j=1:n
        if c1(i,j)<min1
            min1=c1(i,j);
            jmin(i)=j;% position
of min1 on each row
        end
    end
    minrow1(i)=min1;
end;
for i=1:m
```

ໂຮງງານ	ສາງເກັບສິນຄ້າ				
	1	2	3	4	a_i
1	1	2	2	5	500
2	3	5	2	4	300
3	4	1	3	3	600
b_j	400	200	500	300	1,400

```
min2=inf;
for j=1:n
    if j~=jmin(i)
        if c1(i,j)<=min2
            min2=c1(i,j);
        end
    end
end
minrow2(i)=min2;
end
%% Column Difference
mincol1=zeros(1,n);
mincol2=zeros(1,n);
imin=zeros(n,1);
for j=1:n
    minR1=inf;
    for i=1:m
        if c1(i,j)<minR1
            minR1=c1(i,j);
            imin(j)=i;
        end
    end
    mincol1(j)=minR1;
end
for j=1:n
    minR2=inf;
    for i=1:m
```

```

        if i~=imin(j)
            if c1(i,j)<=minR2
                minR2=c1(i,j);
            end
        end
    end
    mincol2(j)=minR2;
end
%% Difference
diffrow=zeros(m,1);
diffcol=zeros(1,n);
for i=1:m
    diffrow(i)=minrow2(i)-
minrow1(i);
end;
for j=1:n
    diffcol(j)=mincol2(j)-
mincol1(j);
end
%% The greatest difference
R=0;
Row=zeros(m,1);
for i=1:m
    if diffrow(i)>=R
        R=diffrow(i);
        iminrow=i;
    end
end
Row(iminrow)=R;
S=0;
Col=zeros(1,n);
for j=1:n
    if diffcol(j)>=S
        S=diffcol(j);
        jmincol=j;
    end
end
Col(jmincol)=S;
great=zeros(1,n);
for j=1:n
    if S>=R
great(jmincol)=Col(jmincol);
        Colline=1;
    else
        great(iminrow)=Row(iminrow);
        Colline=0;
    end
end
%% Search the entry cell
if Colline==1
    j=jmincol;
    R1=inf;
    for i=1:m
        if c1(i,jmincol)<=R1
            R1=c1(i,jmincol);
            igreat=i;
        end
    end

    if
s1(igreat)>d1(jmincol)
        x(igreat,jmincol)=d1(jmincol);

s1(igreat)=s1(igreat)-
d1(jmincol);
        d1(jmincol)=0;
        eliminatorow=0;
    elseif
s1(igreat)<d1(jmincol)
        x(igreat,jmincol)=s1(igreat);

d1(jmincol)=d1(jmincol)-
s1(igreat);
        s1(igreat)=0;
        eliminatorow=1;
    elseif
s1(igreat)==d1(jmincol)
        x(igreat,jmincol)=s1(igreat);
        d1(jmincol)=0;
        s1(igreat)=0;
        eliminatorow=2;
    end
    % Eliminate a column or
a row

```

```

        if elimaterow==0
            for i=1:m
                s1(iminrow)=0;
                elimaterow=1;
            elseif
                s1(iminrow)==d1(jgreat)
                    x(iminrow,jgreat)=s1(iminrow);
                    d1(jgreat)=0;
                    s1(iminrow)=0;
                    elimaterow=2;
                end
            % Eliminate a column
            or a row
                if elimaterow==0
                    for i=1:m
                        c1(i,jmincol)=inf;
                        end
                        for j=1:n
                            c1(i,jgreat)=inf;
                            end
                            elseif elimaterow==1
                                for j=1:n
                                    c1(igreat,j)=inf;
                                    end
                                    end
                                    else % Colline=0;
                                        i=iminrow;
                                        R2=inf;
                                        for j=1:n
                                            if c1(iminrow,j)<R2
                                                R2=c1(iminrow,j);
                                                jgreat=j;
                                            end
                                        end
                                        if
                                            s1(iminrow)>d1(jgreat)
                                                x(iminrow,jgreat)=d1(jgreat);
                                                s1(iminrow)=s1(iminrow)-
                                                d1(jgreat);
                                                d1(jgreat)=0;
                                                elimaterow=0;
                                            elseif
                                                s1(iminrow)<d1(jgreat)
                                                    x(iminrow,jgreat)=s1(iminrow);
                                                    d1(jgreat)=d1(jgreat)-
                                                    s1(iminrow);

```



```

        sums=sums+s1(i);
    end
    sumd=0;
    for j=1:n
        sumd=sumd+d1(j);
    end
    if (sums & sumd) ==0
        return
    end
end

%% The degeneracy
countx=0;
for i=1:m
    for j=1:n
        if x(i,j)>0
            countx=countx+1;
            x1(i,j)=x(i,j);
            x2(i,j)=x(i,j);
        end
    end
end
if countx>=numbasic
    disp('Total cost of non-
degeneracy VAM');
    disp(z);
else
    disp('Total cost of
degeneracy VAM');
    disp(z);
end
end
end

```

ຫລັງຈາກ run ໂຄດ Matlab M-File 3 ຈະໄດ້
ຄໍາຕອບ $2,600 \times 1000 = 260,000$ ກີບ

4. ວິພາກຜົນ

ໃນການສຶກສາຄັ້ງນີ້ ຄະນະຜູ້ຄົ້ນຄວ້າແມ່ນໄດ້ສຶກສາ
ການນໍາໃຊ້ MATLAB ເພື່ອແກ້ໄຂບັນຫາການຂົນສົ່ງ ແຕ່
ໃນການສຶກສາຄັ້ງນີ້ ຄະນະຜູ້ສຶກສາບໍ່ໄດ້ສະເໜີວິທີການ
ຂຽນໂຄດໃນ MATLAB ແກ້ໃນວິທີແບບ MODI ໂດຍ
ການອອກແບບຂອງບັນຫາການຂົນສົ່ງ ສາມາດຂຽນເປັນ
ແບບຈໍາລອງທາງຄະນິດສາດ, ແລ້ວນໍາໃຊ້ MATLAB
ໂຄດໃນ ການຊອກຫາໃຈຜົນ ແລ້ວຈະໄດ້ຜົນໄດ້ຮັບໂດຍມີ
ເປົ້າໝາຍແກ້ບັນຫາຂົນສົ່ງຕາມວິທີ Northwest Corner

Rule, Least Cost Method, Vogel's Approximation Method, and Modi method ເຊິ່ງສອດຄ່ອງກັບ
ທິດສະດີ ໃນສາຂາວິໄຈການດໍາເນີນງານ ຄໍາສະຫຍຸ້ຍ ໄຊ
ຍະລາດ, 2012. ບັນຫາການຂົນສົ່ງເປັນບັນຫາພື້ນຖານ
ຂອງບັນຫາໃນຊີວິດປະຈຳວັນເພາະວ່າມັນກ່ຽວຂ້ອງກັບ
ການຊອກຫາຜົນປະໂຫຍດສູງສຸດ. ສະນັ້ນ, ທ່ານໃດທີ່ມີ
ຄວາມສົນໃຈສາມາດສຶກສາຕໍ່ໂດຍນໍາໃຊ້ການຂຽນໂຄດໃຊ້
ວິທີຂອງ MODI.

5. ສະຫຼຸບຜົນ

ໃນສາຂາວິຊາການວິໄຈດໍາເນີນງານ, ແບບຈໍາລອງ
ທາງຄະນິດສາດຂອງບັນຫາການຂົນສົ່ງເປັນບັນຫາພື້ນຖານ
ຂອງບັນຫາໃນຊີວິດປະຈຳວັນເພາະວ່າມັນກ່ຽວຂ້ອງກັບ
ການຊອກຫາຜົນປະໂຫຍດສູງສຸດ. ໃນໄລຍະຜ່ານມາໄດ້ມີ
ນັກວິໄຈຫຼາຍທ່ານໃຊ້ຄວາມພະຍາຍາມໃນການຊອກຫາວິທີ
ການຕ່າງໆເພື່ອຊອກຫາໃຈຜົນທີ່ດີທີ່ສຸດຂອງບັນຫາການ
ຂົນສົ່ງຊຶ່ງໃນນັ້ນລວມມີ ວິທີ Vogel's Approximation
Method (VAM) ທີ່ເປັນໜຶ່ງວິທີທີ່ດີ ທີ່ສາມາດຊອກຫາ
ຜົນຮັບພື້ນຖານຂຶ້ນຕົ້ນໄດ້ ແຕ່ວ່າເວລາຜ່ານໄປມາຮອດປັດ
ຈຸບັນນີ້ MATLAB ຊຶ່ງເປັນເຄື່ອງມືສໍາຄັນຂອງນັກຄະ
ນິດສາດຫຼາຍທ່ານ ທີ່ໃຊ້ເຂົ້າໃນການແກ້ບັນຫາທາງຄະນິດ
ສາດຫຼາຍຢ່າງເພາະວ່າມັນເປັນເຄື່ອງມືທີ່ມີປະສິດທິຜົນໃນ
ການຊອກຫາໃຈຜົນຂອງບັນຫາ. ໃນບົດວິໄຈນີ້ ຜູ້ຂຽນໄດ້
ຂຽນໂຄດໃນ MATLAB ເພື່ອຊອກຫາຜົນຮັບພື້ນຖານ
ເບື້ອງຕົ້ນຂອງບັນຫາການຂົນສົ່ງໂດຍເນັ້ນໃສ່ 3 ວິທີຂອງ
ການແກ້ບັນຫາການຂົນສົ່ງເຊັ່ນ: ວິທີ Northwest
Corner, ວິທີ Least Cost Matrix, ວິທີ Vogel's
Approximation Method (VAM) ຊຶ່ງການຂຽນໂຄດ
ໃນ MATLAB ຈະໃຫ້ຄວາມສະດວກແກ່ຜູ້ນໍາໃຊ້ໃນກໍ
ລະນີທີ່ມາຕຣິສຂອງບັນຫາການຂົນສົ່ງມີຂະໜາດໃຫຍ່.
ໃນການຂຽນໂຄດຄັ້ງນີ້ປະກອບມີ 2 ປະເພດ ຊຶ່ງ
Matlab M-File 1 ແລະ Matlab M-File
2 ແມ່ນຂຽນໃນຮູບແບບຂອງ script ສະນັ້ນເວລາ run
ແມ່ນ run ໃນໜ້າຕ່າງ Editor ແລ້ວຈະໄດ້ຄໍາຕອບໃນໜ້າ

ຕ່າງ Command Window ສ່ວນ **Matlab M-File 3** ແມ່ນຂຽນໃນຮູບແບບ **function** ຊຶ່ງຫລັງຈາກ run ໃນໜ້າຕ່າງ Editor ແລ້ວຈະໄດ້ບໍ່ຄໍາຕອບໃນໜ້າຕ່າງ Command Window ເລີຍ. ຖ້າຢາກໄດ້ຄໍາຕອບຕ້ອງໄດ້ປ້ອນຂໍ້ມູນໃນໜ້າຕ່າງ Command Window ຕາມຄ່າຂອງ supply, demand, cost matrix, ຂະໜາດຂອງ supply ແລະ ຂະໜາດຂອງ demand ແລ້ວເອີ້ນ [z ,x] = VAM(s, d, c, m, n) ມາໃຊ້ ຈະໄດ້ຄໍາຕອບທີ່ຕ້ອງການ. ຜ່ານການຂຽນໂຄດໃນ 3 ກໍລະນີທີ່ກ່າວມານັ້ນເຫັນວ່າ ວິທີ Northwest Corner ແລະ ວິທີ Least Cost Matrix ແມ່ນບໍ່ຍາກ ແລະ ບໍ່ຍາວປານໃດ ສ່ວນວິທີ Vogel's Approximation Method (VAM) ແມ່ນຂ້ອນຂ້າງຍາກ ແລະ ຍາວຫຼາຍເຕີບ. ໂຄດທັງສາມກໍລະນີນັ້ນສາມາດໃຊ້ໄດ້ຢ່າງມີປະສິດທິຜົນດີ ແລະ ໄດ້ຄໍາຕອບແຜນດຽວກັນກັບໃຊ້ມືແກ້. ສະນັ້ນ, ສະຫລຸບໄດ້ວ່າໃຊ້ MATLAB ເຂົ້າໃນການແກ້ບັນຫາການຂົນສົ່ງແມ່ນງ່າຍ ແລະ ມີປະສິດທິພາບສູງໂດຍສະເພາະໃນກໍລະນີມາຕຣິສຂອງບັນຫານັ້ນໃຫຍ່.

6. ຂໍ້ຂັດແຍ່ງ

ຂ້າພະເຈົ້າໃນນາມຜູ້ຄົນຄວ້າວິທະຍາສາດຂໍປະຕິບານຕົນວ່າ ຂໍ້ມູນທັງໝົດທີ່ມີໃນບົດຄວາມວິຊາການດັ່ງກ່າວນີ້ ແມ່ນບໍ່ມີຂໍ້ຂັດແຍ່ງທາງຜົນປະໂຫຍດກັບພາກສ່ວນໃດ ແລະ ບໍ່ໄດ້ເອື້ອປະໂຫຍດໃຫ້ກັບພາກສ່ວນໃດພາກສ່ວນໜຶ່ງ, ກໍລະນີມີການລະເມີດໃນຮູບການໃດໜຶ່ງ ຂ້າພະເຈົ້າມີຄວາມຍິນດີທີ່ຈະຮັບຜິດຊອບແຕ່ພຽງຜູ້ດຽວ.

7. ເອກະສານອ້າງອີງ

ຄໍາສະຫຍຸຍ ໄຊຍະລາດ. (2012). *ວິໄຈຄຳເນີນງານ*.

ນະຄອນຫຼວງວຽງຈັນ: ມະຫາວິທະຍາໄລແຫ່ງຊາດລາວ.

Alexande, S. (1998). *Theory of Linear and Integer Programming*. John Wiley & sons. Amsterdam: 1998.

Anderson, & Michael Q. (1982). *Quantitative Management Decision Making*, Belmont. England: 1982.

Anderson, M., & T.A, W. (1982). *An Introduction to Management*. England: 1982.

Ansderon, D., Dennis J, S., & Thomas A, W. (2001). *Quantitative*. England.

Bernd, G., & Jiří, M. (2006). *Undersading and Using Linear Programming*, Berlin: Springer. France: 2006.

Bierman, H., Boninni, C., & Hasman, W. (1977). *Quantitative Analysis for Business Decisions (Richard D. Irwin, Inc., Homewood, IL, 1977). 642 pp*. Carifornia: California: Brooks/ Cole Publishing Co.,.

Chames, A., Cooper, W., & Henderson, A. (1953). *An Introduction to Linear Programming*, Wiley, New York.

Dantzig, G. (1963).), “*Linear Programming and Extensions*”, Princeton University Press, Princeton, N J.,

Dantzig, G., & Thapa, M. (1997).). *Linear programming 1: Introduction*. Springer-Verlag. Eglan: 1997.

Dantzig, G., & Thapa, M. (2003). *Linear Programming 2: Theory and Extensions*. Springer-Verlag.

Dunn, A, R., & K.D, R. (1981). *Management Science: A Practical Approach*.

Dykstra, D. (1984). *Mathematical Programming for Natural Resource Management (McGraw-Hill, Inc., New York, 1984). 318 pp*.

Edward, S., & Venkattachalapathy, M. (2011). Modified Vogel's Approximation Method for Fuzzy Transportation Problems” Applied Mathematical Sciences,, *Mathematical Sciences*, Vol.5 (28) 1367-1372.

Hillier, F. (1941).), “*The distribution of a product from several sources to numerouslocalities*”, *J. Math. Phys.* 20,224-230.

Hillier, F. (1995). *Introduction to Operations Research, sixth edition (McGraw-Hill, Inc., New York, 1995). 998 pp*. New York: 1995.