

The Constrained Bottleneck Transportation Problem

Peerayuth Charnsethikul and Saeree Svetasreni
 Operations Research and Management Science Units, Industrial Engineering Department
 Kasetsart University, Bangkok 10903, Thailand

Abstract: Two classes of the bottleneck transportation problem with an additional budget constraint are introduced. An exact approach was proposed to solve both problem classes with proofs of correctness and complexity. Moreover, the approach was extended to solve a class of multi-commodity transportation network with a special case of the multi-period constrained bottleneck assignment problem.

Key words: Bottleneck transportation problem, additional constraint budget

INTRODUCTION

The bottleneck transportation problem with an additional budget constraint referred as the constrained bottleneck transportation problem (CBTP) is an extension of the classical minimum cost flows problem^[1] dealing with finding an appropriated allocation of commodity from a set of suppliers to a group of customers in order to minimize the maximal operation time under supply capacity, demand requirement and budget constraints, Mathematically, the problem can be formulated as two following models.

Model I

Minimize T
 Subject to

$$\begin{aligned}
 T &\geq P_{ij}X_{ij} \quad i = 1, 2, \dots, m, \quad j = 1, 2, \dots, n \\
 \sum_{j=1}^n X_{ij} &= S_i \quad i = 1, 2, \dots, m \\
 \sum_{i=1}^m X_{ij} &= D_j \quad j = 1, 2, \dots, n \\
 \sum_{i=1}^m \sum_{j=1}^n C_{ij}X_{ij} &\leq B \\
 X_{ij} &\geq 0 \quad i = 1, 2, \dots, m, \quad j = 1, 2, \dots, n
 \end{aligned}$$

Model II

Minimize T
 Subject to

$$\begin{aligned}
 T &\geq P_{ij}Y_{ij} \quad i = 1, 2, \dots, m, \quad j = 1, 2, \dots, n \\
 \sum_{j=1}^n X_{ij} &= S_i \quad i = 1, 2, \dots, m \\
 \sum_{i=1}^m X_{ij} &= D_j \quad j = 1, 2, \dots, n
 \end{aligned}$$

$$\begin{aligned}
 \sum_{i=1}^m \sum_{j=1}^n C_{ij}X_{ij} &\leq B \\
 MY_{ij} - X_{ij} &\geq 0 \quad i = 1, 2, \dots, m, \quad j = 1, 2, \dots, n \\
 X_{ij} &\geq 0, \quad Y_{ij} = \{0,1\} \quad i = 1, 2, \dots, m, \quad j = 1, 2, \dots, n
 \end{aligned}$$

From the above models, the following terms can be defined;

- * m and n represent the number of suppliers and customers, respectively,
- * X_{ij} represents the amount of commodities transported from supplier i to customer j,
- * P_{ij} represents the transportation time per unit of X_{ij} in case of model I or represents the total transportation time regardless of X_{ij} in case of model II,
- * Y_{ij} represents the 0-1 variable related to X_{ij} that if $X_{ij} \geq 0$ then $Y_{ij} = 1$ otherwise $Y_{ij} = 0$
- * S_i represents the capacity of supplier i,
- * D_i represents the requirements of customer j,
- * M represents a large positive number with at least equal to the minimum between the maximal capacity among suppliers and the maximal requirement among customers,
- * T represents the amount with at least equal to the maximal total transportation time,
- * C_{ij} represents the transportation cost per unit of X_{ij}
- * B represents the upper bound of total transportation cost.

The major difference between both models is the representation of T. In the first model, T is proportional to X_{ij} with a constant P_{ij} while T is equal either P_{ij} or 0, otherwise. The zero-one variable, Y_{ij} is introduced in the second model related with X_{ij} in order to identify T.

Corresponding Author: P. Charnsethikul, Operations Research and Management Science Units, Industrial Engineering Department, Kasetsart University, Bangkok 10903, Thailand

Step 3: If $Z^* > B$ then let $T^* = \text{Min}\{P_{ij}, \forall P_{ij} > T^*\}$ and repeat step 2 again. Otherwise, the last obtained T^* and X_{ij} is an optimal solution.

The validity of algorithm II is obvious since the obtained T^* from step 1 represents a lower bound on T . Step 2-3 is used to check and identify whether there exists a solution under budget B or not leading to the fact that the first feasible solution detected has the optimal T as desired. Nevertheless, a key computational factor in this algorithm deals with solving the bottleneck transportation problem in the first step. A recommended algorithm is proposed as an exercise in Ahuja et al.^[3]. The procedure starts searching for a feasible solution under an initial maximum P_{ij} for all

$X_{ij} > 0$ by solving a transportation problem with prohibition on upper-bound allocation of X_{ij} with P_{ij} larger than the initial maximal value. If no solution is detected, the next smallest P_{ij} which is larger than the previous assigned, will be set as the new maximum. Then, continue in the similar steps until a first feasible solution is detected and terminated as an optimal solution. Next, complexity of algorithm will be analyzed under the following theorem.

Theorem 3: Algorithm II can be bounded within $O(m^4 n^4 \text{Log}(n+m))$

Proof: Similar to algorithm I, step 0 of algorithm II can be bounded within $O(m^3 n^3 \text{Log}(n+m))$ using the dual simplex network algorithm. In step 1, a sequence of transportation problems is solved at most no more than mn times due to alternating all possible P_{ij} for all i, j . Therefore, step 1 has a complexity of $O(m^4 n^4 \text{Log}(n+m))$ which is similar to the case of repeating steps 2 – 3. Hence, the theorem is as stated.

A special case of both models I and II is when $m = n$, S_i and D_j for all i and j are equal to 1 and X_{ij} is restricted to either 0 or 1. Both models become the same problem of minimizing the maximal P_{ij} under the assignment constraints and an additional constraint. This problem referred as “The Constrained Mini-Max Linear Assignment Problem” can be solved within $O(n^5)$ ^[4].

Extensions to a multi-commodity case:

The algorithms proposed in the previous section can be extended to solve two classes of multi-commodity transportation network represented as following mathematical programming models.

Model III

Minimize T
Subject to

$$\begin{aligned} T &\geq P_{ijk} X_{ijk} & i = 1, 2, \dots, m, \\ & & j = 1, 2, \dots, n, \\ & & k = 1, 2, \dots, s \\ \sum_{j=1}^n X_{ijk} &= S_{ik} & i = 1, 2, \dots, m, \\ & & k = 1, 2, \dots, s \\ \sum_{i=1}^m X_{ijk} &= D_{jk} & j = 1, 2, \dots, n, \\ & & k = 1, 2, \dots, s \\ \sum_{i=1}^m \sum_{j=1}^n \sum_{k=1}^s C_{ijk} X_{ijk} &\leq B \\ X_{ijk} &\geq 0 & i = 1, 2, \dots, m, \\ & & j = 1, 2, \dots, n, \\ & & k = 1, 2, \dots, s \end{aligned}$$

Model IV

Minimize T
Subject to

$$\begin{aligned} T &\geq P_{ijk} Y_{ijk} & i = 1, 2, \dots, m, \\ & & j = 1, 2, \dots, n, \\ & & k = 1, 2, \dots, s \\ \sum_{j=1}^n X_{ijk} &= S_{ik} & i = 1, 2, \dots, m, \\ & & k = 1, 2, \dots, s \\ \sum_{i=1}^m X_{ijk} &= D_{jk} & j = 1, 2, \dots, n, \\ & & k = 1, 2, \dots, s \\ \sum_{i=1}^m \sum_{j=1}^n \sum_{k=1}^s C_{ijk} X_{ijk} &\leq B \\ MY_{ijk} - X_{ijk} &\geq 0 & i = 1, 2, \dots, m, \\ & & j = 1, 2, \dots, n, \\ & & k = 1, 2, \dots, s \\ X_{ijk} \geq 0, Y_{ijk} &= \{0, 1\} & i = 1, 2, \dots, m, \\ & & j = 1, 2, \dots, n, \\ & & k = 1, 2, \dots, s \end{aligned}$$

The index k represents the order of commodity up to the total number of s . In both cases, algorithms I and II from the previous section can be adapted to solve respectively and can be summarized as following algorithm III and algorithm IV.

Algorithm III

Step 0: Solve the following transportation problems for $k = 1, 2, \dots, s$.

$$\begin{aligned} \text{Minimize} & \sum_{i=1}^m \sum_{j=1}^n C_{ijk} X_{ijk} = Z_k^* \\ \text{Subject to} & \\ \sum_{j=1}^n X_{ijk} &= S_i & i = 1, 2, \dots, m \\ \sum_{i=1}^m X_{ijk} &= D_j & j = 1, 2, \dots, n \\ X_{ijk} &\geq 0 & i = 1, 2, \dots, m, \\ & & j = 1, 2, \dots, n \end{aligned}$$

Then, check whether $Z^* = \sum_{k=1}^s Z_k^*$ is less than or equal to B or not. If yes then proceeds step 1 Otherwise,

there is no feasible solution. Let $T_0 = \text{Max}_{\forall ijk} \{P_{ijk} X_{ijk}\}$, $T_u = T_0$ and $T_1 = 0$.

Step 1: Then, let $T^* = (T_u + T_1) / 2$ and solve s transportation problems from step 0 with bounded variable constraints as follows.

$$0 \leq X_{ijk} \leq T^*/P_{ijk} \quad \forall i, j$$

Step 2: Check whether Z^* is less than or equal B or not. If yes then let $T_u = T^*$ and proceed to step 3. Otherwise, let $T_1 = T^*$ and repeat steps 1-2 again.

Step 3: Check if $T_u - T_1$ is less than or equal to ϵ , then the last obtained solution from step 1 is an optimal solution with T^* under tolerance ϵ . If not, repeat steps 1-2 again.

Algorithm IV

Step 0: Solve the transportation problem as described in step 0 of algorithm I. Then, check whether $Z^* < B$ or not. If yes then proceeds step 1. Otherwise, there is no feasible solution.

Step 1: Solve the following bottleneck transportation problems for $k = 1, 2, \dots, s$.

$$\begin{aligned} \text{Minimize} \quad & T = T_k^* \\ \text{Subject to} \quad & T \geq P_{ijk} Y_{ijk} \quad i = 1, 2, \dots, m, \\ & \quad \quad \quad j = 1, 2, \dots, n \\ & \sum_{j=1}^n X_{ijk} = S_{ik} \quad i = 1, 2, \dots, m \\ & \sum_{i=1}^m X_{ijk} = D_{jk} \quad j = 1, 2, \dots, n \\ & MY_{ijk} - X_{ijk} \geq 0 \quad i = 1, 2, \dots, m, \\ & \quad \quad \quad j = 1, 2, \dots, n \\ & X_{ijk} \geq 0, Y_{ijk} = \{0, 1\} \quad i = 1, 2, \dots, m, \\ & \quad \quad \quad j = 1, 2, \dots, n \end{aligned}$$

Let $T^* = \text{Min}_{\forall k} \{T_k^*\}$

Step 2: Let $C^*_{ijk} = \infty$ if $P_{ijk} \geq T^*$, otherwise, set $C^*_{ijk} = C_{ijk}, \forall i, j, k$ and then, solve the transportation problems for all possible k with the objective function,

$$\text{Minimize} \quad \sum_{i=1}^m \sum_{j=1}^n C_{ijk} X_{ijk} = Z_k^*$$

Step 3: If $Z^* > B$ then let $T^* = \text{Min}\{P_{ijk}, \forall P_{ijk} > T^*\}$ and repeat step 2 again. Otherwise, the last obtained T^* and X_{ijk} is an optimal solution.

Correctness and validity of both algorithms III and IV are obvious since algorithm III is terminated with a feasible solution at convergence of T^* within ϵ similar to previous stated Theorem I and algorithm IV is terminated with the first detected feasible solution with T^* as the optimal value. Complexities of both algorithms are just multiplications of s to the complexity of algorithms I and II respectively.

Again, the special structure in this case is when $m = n$, S_{ik} and D_{jk} , for all possible i, j, k are 1 and all X_{ijk} are either 0 or 1 leading to the same problem statement of both models III and IV. If index k represents time period no., the problem becomes the multi-period mini-max constrained linear assignment problem by which can be solved using algorithm IV with modifications on applying algorithms for the linear sum assignment problem to solve the corresponding transportation problem in steps 0 and 2 and algorithms for the bottleneck assignment problem to solve the corresponding bottleneck transportation problem in step 1. These algorithms are summarized and can be found in Burkard and Derigs^[5]. It is well known that complexity of both linear sum and mini-max assignment problem is $O(n^3)$. Therefore, complexity of algorithm IV for this special case is within $O(n^5 s^2)$ since the procedure bottleneck is to the iteration of steps 2 and 3 at most $n^2 s$ times with solving the assignment problems s times in each iteration.

CONCLUSION

Two classes of the constrained bottleneck transportation problem have been defined and solved by a proposed approach within reasonable worst case times. The approach can also be extended efficiently and effectively to solve a class of multi-commodity problem including the special case with the assignment constraints.

REFERENCES

1. Ford and Fulkerson, 1962. Flows in Network. Princeton University Press, New Jersey.
2. Plotzkin and Tardos, 1990. Improved dual network simplex. Proc. 1st ACM- SIAM Symposium on Discrete Algorithm, pp: 367-376.
3. Ahuja, R.K., T.L. Magnanti and J.B. Orlin, 1993. Network Flows Theory, Algorithms and applications. Prentice Hall.
4. Charnsethikul, P., 2000. The constrained mini-max linear assignment problem, Optimization Methods and Software, pp: 37-48.
5. Burkard, R.E. and U. Derigs, Assignment and Matching Problems: Solution Methods with Fortran Programs. Lecture Notes in Economics and Mathematical Systems, Vol. 184, Springer-Verlag, Berlin, Heidelberg.