

A novel approach for solving a variant of Transportation Problem

Swapnil Kumar¹, Sasikanth Goteti²

¹swapnil.bit@iitbombay.org

²raghavas@thoughtworks.com

Abstract

In this article we consider a certain sub class of Integer Equal Flow problem, which are known NP hard [8]. Currently there exist no direct solutions for the same. It is a common problem in various inventory management systems. Here we discuss a local minima solution which uses projection of the convex spaces to resolve the equal flows and turn the problem into a known linear integer programming or constraint satisfaction problem which have reasonable known solutions and can be effectively solved using simplex or other standard optimization strategies.

Keywords— Integer Equal Flow, Transportation, Constraint Satisfaction

1 Problem Space

Integer equal flow problems are known to be NP-Hard as observed by Meyers and Schulz [8]. Most solutions to this would require graph theoretic language to formulate it correctly, as proposed by Morrison et.al [9]. Effective algorithms like network simplex can be used to iteratively improve upon a simple feasible solution. We can formally define an integer equal flow problem as:

$$\begin{aligned} \min_X \quad & c^T X \\ \text{s.t.} \quad & \sum_{j:(i,j) \in A} x_{ij} - \sum_{j:(j,i) \in A} x_{ij} = b_i \\ & x_{ij} \geq 0 \quad \forall j \in N \\ & x_{ij} \leq u_{ij} \\ & x_{ik} = t \quad \forall ik \in Class(t) \end{aligned} \quad (1)$$

One could extend the same definition of network flow optimization to specific cases of transportation and assignment problems. An equal transportation problem can be

extended as:

$$\begin{aligned}
 \min_X \quad & c^T X \\
 \text{s.t.} \quad & \sum_{j=1}^n x_{ij} = b_i \\
 & \sum_{i=1}^n x_{ij} = a_j \\
 & x_{ij} \geq 0 \\
 & x_{ij} \leq u_{ij} \\
 & x_{ik} = t \quad \forall ik \in \text{Class}(t)
 \end{aligned} \tag{2}$$

In many cases of transportation or assignment problems, it is important to follow the same routes or same inventory allocations over a period of repeated inventory assignment cycles mainly to reduce maintenance costs or other taxation charges. To resolve these scenarios, it is important that the same inventory takes the same route every time. So if we add a third index τ for time period, we can define an equal assignment over time. We define a corresponding same route transportation problem as:

$$\begin{aligned}
 \min_X \quad & c^T X \\
 \text{s.t.} \quad & \sum_{j=1}^n x_{ij\tau} = b_{i\tau} \\
 & \sum_{i=1}^n x_{ij\tau} = a_j \\
 & x_{ij\tau} \geq 0 \\
 & x_{ij\tau} \leq u_{ij\tau} \\
 & x_{ik\tau} = t_{ik} \quad \forall \tau \in \text{Class}(t_{ik})
 \end{aligned} \tag{3}$$

Definition 1.1. *Class* We define the set of all arcs in a same transportation with same value and same indexes as $\text{Class}(t_{ik\tau})$ where indices are represented as usual interpretation

Example 1.1.1. $\text{Class}(t_{i,,})$ represents set of all source vectors that are equal to t_i

Example 1.1.2. $\text{Class}(t_{,k,})$ represents set of all destination vectors that are equal to t_k

Example 1.1.3. $\text{Class}(t_{i,k,})$ represents set of all arc scalars that are equal to t_{ik} across all τ

Example 1.1.4. $\text{Class}(t_{,,})$ represents set of all transportation problems that have same solution space across all τ

Often in usage we will ignore the , and leave it for interpretation with the subscript used. One can extend the same problem to a much broader setting of assignment problem a typical same inventory assignment problem would look like:

$$\begin{aligned}
 \min_X \quad & c^T X \\
 \text{s.t.} \quad & \sum_{j=1}^n x_{ij\tau} = 1 \\
 & \sum_{i=1}^n x_{ij\tau} = 1 \\
 & x_{ij\tau} \in \{0, 1\} \\
 & x_{ik\tau} = t_{ik} \quad \forall \tau \in \text{Class}(t_{ik})
 \end{aligned} \tag{4}$$

One could formulate generalized assignment and generalized transportation problems under the same breadth. We will specifically deal with a generalized same route assignment problem, which can be formulated as:

$$\begin{aligned}
 \min_X \quad & c^T X \\
 \text{s.t.} \quad & \sum_{j=1}^n x_{ij\tau} = b_{i\tau} \\
 & \sum_{i=1}^n x_{ij\tau} = 1 \\
 & x_{ij\tau} \in \{0, 1\} \\
 & x_{ik\tau} = t_{ik} \quad \forall \tau \in \text{Class}(t_{ik})
 \end{aligned} \tag{5}$$

2 Feasibility Certificates

Feasibility certificate or a gale certificate [4] typically can be understood as determining if the constraints in a network optimization have a feasible flow or not. This is a very easy to determine in a simple transportation problem - if its balanced you are always guaranteed to have a feasible solution. One can quickly verify it using a simple north-west corner solution. For a more rigorous treatment of the same we will need a Matroid theory. In-specific to solving a forbidden arc transportation problem we will need to prove the existence of a Monge sequence as observed by Shamir et al. [1]. We can define a forbidden-arc same route assignment problem as below.

Definition 2.1. We call the set of all arcs where the flow is constrained to void as forbidden arcs and represent this set by \mathfrak{F}

$$\mathfrak{F} = \{(i, k, \tau) \mid x_{ik\tau} = 0\}$$

$$\begin{aligned}
 \min_X \quad & c^T X \\
 \text{s.t.} \quad & \sum_{j=1}^n x_{ij\tau} = b_{i\tau} \\
 & \sum_{i=1}^n x_{ij\tau} = 1 \\
 & x_{ij\tau} \in \{0, 1\} \\
 & x_{ik\tau} = t_{ik} \quad \forall \tau \in \text{Class}(t_{ik}) \\
 & x_{ij\lambda} = 0 \quad \forall (i, j, \lambda) \in \mathfrak{F}
 \end{aligned} \tag{6}$$

where \mathfrak{F} is the set of all forbidden arcs (note that the equal flow arcs are counted multiple times in the forbidden arc constraints \mathfrak{F} as they are equally 0 every where).

In a similar setting one could define the feasibility certificate problem of a forbidden-arc same route generalized assignment problem as.

$$\begin{aligned}
 \exists \quad & X \\
 \text{s.t.} \quad & \sum_{j=1}^n x_{ij\tau} = b_{i\tau} \\
 & \sum_{i=1}^n x_{ij\tau} = 1 \\
 & x_{ij\tau} \in \{0, 1\} \\
 & x_{ik\tau} = t_{ik} \quad \forall \tau \in \text{Class}(t_{ik}) \\
 & x_{ij\lambda} = 0 \quad \forall (i, j, \lambda) \in \mathfrak{F}
 \end{aligned} \tag{7}$$

3 Problem reformulations

3.1 Stacking By index

One could reformulate the problem by stacking the individual transportation problems into one big transportation problem and setting or forcing all the irrelevant arcs forcibly to zero. Once this is done the transportation matrix would look like a diagonal matrix with indexed transportation problems along the diagonal and the entire diagonal matrix can itself be handled as a forbidden arc transportation problem.

A similar transportation matrix would look like

$$\begin{array}{cccccc} x_{000} & x_{010} & 0 & 0 & \cdots & 0 \\ x_{100} & x_{110} & 0 & 0 & \cdots & 0 \\ 0 & 0 & x_{001} & x_{011} & \cdots & 0 \\ 0 & 0 & x_{101} & x_{111} & \ddots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \ddots & x_{01\tau} \\ 0 & 0 & 0 & \cdots & x_{n0\tau} & x_{n1\tau} \end{array}$$

3.2 Resolving Sparsity

A diagonal transportation matrix like the one above has serious sparsity problems making it difficult to solve it using conventional transportation strategies. Hence we will modify the cost function to throw high cost to get rid of the sparse arcs or force set them to 0s, so an equivalent form of a sparse transportation problem can be reinterpreted as:

$$\begin{array}{ll} \min_X & c^T X + \Lambda^T Y \quad \forall y_{(i,j,\lambda)} \in \mathfrak{F} \\ \text{s.t.} & \sum_{j=1}^n x_{ij} = b_i \\ & \sum_{i=1}^n x_{ij} = 1 \\ & x_{ij} \in \{0, 1\} \\ & x_{ik} = t_{ik} \quad \forall \tau \in \text{Class}(t_{ik}) \\ & \lambda \gg 1 \end{array} \quad (8)$$

Note that we have converted the sparsity constraints into the objective by taking the dual form for those particular constraints.

3.3 Resolving equality constraints

One could further the same idea and even get rid of the equality conditions but at the cost of introducing a quadratic objective. One could reformulate the problem as

$$\begin{array}{ll} \min_X & c^T X + \Lambda^T Y + \lambda \sum_{(i,k) \in \text{Class}(t_k)} (x_{ik} - t_{ik})^2 \quad \forall y_{(i,j,\lambda)} \in \mathfrak{F} \\ \text{s.t.} & \sum_{j=1}^n x_{ij} = b_i \\ & \sum_{i=1}^n x_{ij} = 1 \\ & x_{ij} \in \{0, 1\} \\ & \lambda \gg 1 \end{array} \quad (9)$$

The above formulation is a standard generalized quadratic transportation problem which has some known approaches. One could use an equalization method as proposed by Marcin et al .[2] But this is still a quadratic Transportation problem with no bounds on convergence that can be given.

One could even stack the indexes in a multidimensional way and arrive a 3D transportation problem which has some known solutions as proposed by stefan et al . [10] But even this approach cannot get rid of the quadratic cost function. One could have easy solutions to this problem once it has been linearized in some for as noted by Bertsekas et al [11] . But it is not possible to linearize a quadratic transportation problem the same way it could be done with an assignment problem with Glover Linearization [5]. So its significantly harder to solve a quadratic transportation problem than even solving a quadratic assignment problem and the state of the art solutions for QAP almost never scale over $n = 20$ [3]. In case of quadratic transportation problem its much harder to even formulate the complexity as a dependent of n .

4 Exploiting the inherent symmetries

Definition 4.1. *Similar Route: Similar routes in a same route transportation problem are the row vector which are bounded by the equality constraint. Its represented by μ_k . It is simply the arcs represented by indices in $Class(t_i)$*

$$\mu_k = \{x_{ik\tau} \mid (i, \tau) \in Class(t_k)\}$$

Definition 4.2. *Equibounded: A same route transportation problem is called equibounded if all the arcs in the similar routes have the same bounds, A same route transportation problem is equibounded if it satisfies the predicate:*

$$\exists l_{ik}, \exists u_{ik} (l_{ik} < t_{ik} < u_{ik}, \forall x_{ik\tau} \in A_{ik} \in A_k \mid x_{ik\tau} = t_{ik})$$

Definition 4.3. *Symbol of A transportation: The span of a same route transportation problem is the count of its sources, destinations and similarity classes. It is represented by $\varsigma_{ik\tau}$ it can be assigned a value $i * k * \tau$ called the size of a transportation, in general a transportation problem can be represented by $\varsigma_{ik\tau}$ we say a transportation problem is feasible by its symbol directly*

Definition 4.4. *Symbol of An Assignment: The span of a same route Assignment problem is the count of its sources, destinations and similarity classes. It is represented by $\alpha_{ik\tau}$ it can be assigned a value $i * k * \tau$ called the size of a Assignment, in general an Assignment problem can be represented by $\alpha_{ik\tau}$, we say a assignment problem is feasible by its symbol directly.*

4.1 Equivalence Theorems

Theorem 1 (Weak Equivalence). *For every same route Assignment problem with a feasibility certificate there exists a corresponding Same route transportation problem with a feasibility certificate whenever the problem is equibounded*

Proof. Its very easy to see that for every transportation setup there exists an equivalent assignment problem that can be solved but to prove the equivalence we need to prove the converse as well.

Since the transportation problem and its equivalent Assignment problem are equibounded we will assume that the lower bound is 0 and just prove for the case of upper bound , similar proof can be extended to the case of Upper bound as well.

We conduct the proof by induction:

for $n=1$, if we have $x_{0i} + x_{0(i+1)} + \dots = b_i$, wherever we have $x_{ij} > 0$ the corresponding assignment variables should all be set to 1 else 0. if the assignment problem is surplus then we can randomly sample any set of variables. If its sub-surplus then the transportation problem would have never given a feasibility certificate.

if for some $n=k$, $\varsigma_{ik\tau} \iff \alpha_{ik\tau} \wedge \varsigma_{i(k+1)\tau}$ then we need to prove that $\alpha_{i(k+1)\tau}$

For the $(k+1)^{th}$ route take the $x_{i(k+1)}$ values from $\varsigma_{i(k+1)\tau}$ and randomly assign 1's to corresponding variables and subtract these assigned values from both $\varsigma_{i(k+1)\tau}$ and $\alpha_{i(k+1)\tau}$, the remaining constraints correspondingly constitute a $\varsigma_{i(k)\tau}$ and $\alpha_{i(k)\tau}$ which we know are feasible by inductive step definition.

Intuitively we can reduce every $\alpha_{i(k+1)\tau}$ into a $\alpha_{i(k)\tau}$ and $\alpha_{i(1)\tau}$ both of which we know are feasible . \square

Theorem 2 (Strong Equivalence). *For every same route Assignment problem with a feasibility certificate there exists a corresponding Same route transportation problem with a feasibility certificate*

Proof. In case of unequibound we split the variables into two and realize that the equivalent problem is solvable for every variable in $\varsigma_{ik\tau}$ we just need to realize that there exists a $\varsigma_{i(k+k')\tau} \mid \varsigma_{ik\tau} \subset \varsigma_{i(k+k')\tau} \wedge \varsigma_{ik\tau}$ is equibound. We just need to reformulate a corresponding $\alpha_{i(k+k')\tau}$ and we are done , proof left to the reader. one could also use direct induction like in equibound case. you can refer to 8.3 for a more detailed explanation. \square

5 Category Theory

Definition 5.1 (Covariant Functor). *If \mathfrak{A} and \mathfrak{B} are two categories then a functor between them is a map that carries every arrow of the category \mathfrak{A} to an arrow in \mathfrak{B} between the same two objects such that identity and composition are preserved.*

Example 5.1.1. *Category of open sets in a topology and boolean algebras representing the inclusion the base sets.*

Definition 5.2 (Contravariant Functor). *If \mathfrak{A} and \mathfrak{B} are two categories then a functor between them is a map that carries every arrow of the category \mathfrak{A} to an arrow in \mathfrak{B} between the same two objects such that identity and composition are preserved and the direction of the arrows are reversed*

Example 5.2.1. *Presheafs in geometry*

Example 5.2.2. *category of convex sets and category of sets of linear in-equations can have a contravariant functor that represents the set of all linear equations that contain the given hull. Inclusions are reversed*

Definition 5.3 (Natural transformation). [7] *Natural transformation is a morphism that carries every arrow in the on the functors naturally such that the below diagram commutes.*

$$\begin{array}{ccc} F(X) & \xrightarrow{F(f)} & F(Y) \\ \downarrow \eta_x & & \downarrow \eta_y \\ G(X) & \xrightarrow{G(f)} & G(Y) \end{array}$$

6 Elimination Strategies

Fourier-Motzkin Elimination (FME) is one of the most common ways to assign feasibility certificates through variable elimination. However, it has limited practicality due to its double-exponential worst time complexity. Even the parallel deployment of FME can only linearly speed-up the process, for both dense and sparse problems. One could also eliminate variables by realizing that for binary constraints you would never have more than 2^n constraints. Unlike in the case of Fourier-Motzkin where coefficients can be other than 0 or 1. So most constraints are going to be duplicates or just inclusions and the weaker constraints can be eliminated straight away. In fact one can rigorously prove that the constraint space would form a measure with values as integers. But for now we will omit that and just propose the algorithm for generating the constraints in the section below which is all that is needed for the purpose of this article. We call this Elimination strategy Fourier-Binary-Constraint-Elimination(FBCE). [6].

7 Solution Approaches

Theorem 3. *Variable elimination is a natural transformation of projection between convex spaces*

Proof. Its clear to see that every set of linear equations can be mapped to a convex space by a contravariant functor. Every elimination morphism is an inclusion map in the category of convex sets hence it must have a contravariant inclusion. Simply put if a convex set contains another then its corresponding linear algebraic set must also have a morphism. The following diagram commutes, where h is the inclusion map on the set of linear equations (a kin to presheaf in algebraic geometry). \square

$$\begin{array}{ccc} \text{lin}(X) & \xrightarrow{\text{lin}(h)} & \text{lin}(Y) \\ \downarrow \eta_x & & \downarrow \eta_y \\ \text{plin}(X) & \xrightarrow{\text{plin}(h)} & \text{plin}(Y) \end{array}$$

Corollary 4 (Projection). *For every s_{ik} there exists a projection in $\mathcal{O}(2^i)$ constraints and k variables*

Proof. Elimination strategy FBCE (6) is a projection on the convex space, hence the proof follows from above theorem \square

Corollary 5 (Intersection). *For every $\varsigma_{ik\tau}$ there exists a corresponding constraint satisfaction problem in $\mathcal{O}(2^i)$ constraints and k variables*

Proof. trivial once we have established the natural transformation, all inclusions are trivial. this is similar to gluing axiom on sheaves only in case of linear equation its trivial. \square

8 Case Study: Car Rental

Similar variant of transportation problem is used to solve the feasibility of different types of requests that a car rental service provider can simultaneously satisfy.

In general, the service provider owns a fleet of cars belonging to different models (brand/build). Since, at least few cars are always under maintenance/service, number of cars available for renting varies over days for every model. We represent m_{db} as the number of cars of model b available on d^{th} day. Further B represents list of all models which the service provider owns, so, $b \in B$.

A typical request is represented as $r_i(d_i, \tau_i, j_i, n_i)$ where, r_i is the request of i^{th} customer; d_i and τ_i are the starting day of the request and the number of days for which cars are to be rented such that $1 \leq d_i + \tau_i - 1 \leq T$; j_i is models of cars requested such that $j_i \subset B$; and, n_i is the total number of cars requested. Also, let's say R is the set of all the requests $r_i(d_i, \tau_i, j_i, n_i)$ and T is the total Time Period under consideration. Finally, let's say R_d is the set of all the request which spans across d^{th} day.

Case 8.1. *No car is under maintenance and cars can be rented for only one day*

We begin with a simplified scenario where no car is under maintenance at any point in time. Hence, m_{db} can be simply written as m_b . Further in this case, we assume that cars are rented for just one day, making $\tau_i = 1$ for all requests. We would lift these restrictions in subsequent sections.

Since this scenario doesn't have requests spanning over multiple days, it doesn't require equal flow analysis and can be solved independently for every day. Let's say x_{ibd} is the number of cars allocated to i^{th} request from brand b on d^{th} day. So, all the requests can be accepted iff,

$$\begin{aligned} & \exists \quad X \\ \text{s.t.} \quad & \sum_{b \in B} x_{ibd} \geq n_i \quad \forall i \in R_d \\ & x_{ibd} = 0 \quad \forall (i, b, d) \in \mathfrak{F} \\ & \sum_{i \in R} x_{ibd} \leq m_b \quad \forall b \in B \\ & x_{ibd} \geq 0 \end{aligned} \tag{10}$$

Here second equation represents the forbidden arcs \mathfrak{F} introduced in section 2 and are defined here as:

$$\mathfrak{F} = \{(i, b, d) \mid i \in R_d; b \notin j_i\}$$

Case 8.2. *No car is under maintenance and cars can be rented for multiple consecutive days*

Let's say x_{ibd} is the number of cars allocated to i^{th} request from brand b on d^{th} day, such that, $d_i \leq d \leq d_i + \tau_i - 1$. In this scenario, we need to establish equal flow constraints

as same set of cars need to be allocated across all requested days for a particular request. We can use $Class(x_{ib})$, introduced in section 1, to represent all the arcs (i, b, d) that are equal to x_{ib} across all d , such that, $d_i \leq d \leq d_i + \tau_i - 1$. Hence, all requests can be accepted iff,

$$\begin{aligned} & \exists \quad X \\ \text{s.t.} \quad & \sum_{b \in B} x_{ibd} \geq n_i \quad \forall i \in R; d: d_i \leq d \leq d_i + \tau_i - 1 \\ & x_{ibd} = 0 \quad \forall (i, b, d) \in \mathfrak{F} \\ & \sum_{i \in R_d} x_{ibd} \leq m_b \quad \forall b \in B; d: 1 \leq d \leq T \\ & x_{ibd} = x_{ib} \quad \forall (i, b, d) \in Class(x_{ib}) \\ & x_{ibd} \geq 0 \end{aligned} \quad (11)$$

Here forbidden arcs \mathfrak{F} are defined as:

$$\mathfrak{F} = \{(i, b, d) \mid i \in R; \text{ either } b \notin j_i \text{ or } d \notin [d_i, d_i + \tau_i - 1]\}$$

Case 8.3. Some cars are under maintenance and cars can be rented for multiple consecutive days

In the most generalized scenario, we take the case where m_b is not constant over days and hence needs to be represented as m_{bd} to reflect the supply of brand b on d^{th} day. Since same set of cars are supposed to be allocated to a particular request across all the requested days, replacing m_b with m_{bd} in equation 11 doesn't solve the problem directly. Let's say cars with serial number S1 and S2 are available on day 1 and cars with serial number S2 and S3 are available on day 2. Although, in this case, $m_{bd} = 2$ for $d = 1, 2$, there is only 1 car with serial number S2 that can be given to a request asking for cars on both days.

This *unequibound* problem is solved by using *variable splitting* method described in section 4.1. There can be a set of cars for every brand/build which are available throughout T , so their numbers can be represented as m_{bT} . Number of cars of a particular model b which are available on day d but are not available throughout T can be written as $m_{bd'}$. Hence we have:

$$m_{bT} + m_{bd'} = m_{bd}$$

Inventories which are available for a set of consecutive days d such that $d_i \leq d \leq d_i + \tau_i - 1$, but are not available throughout T , are given by $m_{bd_i\tau_i}$. It should be noted that $m_{bd_i\tau_i} \leq m_{bd'}$ where $d_i \leq d \leq d_i + \tau_i - 1$.

Finally, we also split the variable x_{ibd} into x_{ibd}^+ and x_{ibd}^- , where x_{ibd}^+ is supplied from common pool of m_{bT} , whereas x_{ibd}^- is supplied from $m_{bd_i\tau_i}$. Here all requests are accepted iff,

$$\begin{aligned}
 & \exists \quad X \\
 \text{s.t.} \quad & \sum_{b \in B} x_{ibd}^+ + x_{ibd}^- \geq n_i & \forall i \in R; d : d_i \leq d \leq d_i + \tau_i - 1 \\
 & x_{ibd}^+, x_{ibd}^- = 0 & \forall (i, b, d) \in \mathfrak{F} \\
 & \sum_{i \in R_d} x_{ibd}^+ \leq m_{bT} & \forall b \in B; d : 1 \leq d \leq T \\
 & \sum_{i \in R_d} x_{ibd}^- \leq m_{bd'} & \forall b \in B; d : 1 \leq d \leq T \\
 & x_{ibd}^- \leq m_{bd_i \tau_i} & \forall i \in R; b \in B; d : d_i \leq d \leq d_i + \tau_i - 1 \\
 & x_{ibd}^+ = x_{ib}^+ & \forall (i, b, d) \in \text{Class}(x_{ib}^+) \\
 & x_{ibd}^- = x_{ib}^- & \forall (i, b, d) \in \text{Class}(x_{ib}^-) \\
 & x_{ibd}^+, x_{ibd}^- \geq 0
 \end{aligned} \tag{12}$$

Again forbidden arcs \mathfrak{F} are defined as earlier:

$$\mathfrak{F} = \{(i, b, d) \mid i \in R; \text{ either } b \notin j_i \text{ or } d \notin [d_i, d_i + \tau_i - 1]\}$$

References

- [1] Ilan Adler, Alan J. Hoffman, and Ron Shamir. Monge and feasibility sequences in general flow problems. *Discrete Applied Mathematics*, 44(1):21 – 38, 1993.
- [2] Marcin Anholcer. The nonlinear generalized transportation problem with convex costs. In *Discrete Applied Mathematics*, 2015.
- [3] R.E. Burkard, S. Karisch, and F. Rendl. Qaplib-a quadratic assignment problem library. *European Journal of Operational Research*, 55(1):115 – 119, 1991.
- [4] David Gale. A theorem on flows in networks. *Pacific J. Math.*, 7(2):1073–1082, 1957.
- [5] Serigne Gueye and Philippe Michelon. A linearization framework for unconstrained quadratic (0-1) problems. *Discrete Applied Mathematics*, 157(6):1255 – 1266, 2009. Reformulation Techniques and Mathematical Programming.
- [6] Christoph W. Keler. Parallel fourier-motzkin elimination. *Boug L., Fraigniaud P., Mignotte A., Robert Y. (eds) Euro-Par'96 Parallel Processing. Euro-Par 1996. Lecture Notes in Computer Science, vol 1124. Springer, Berlin, Heidelberg*, 1124:66–71, 1996.
- [7] Saunders MacLane. *Categories for the Working Mathematician*. Springer-Verlag, New York, 1971. Graduate Texts in Mathematics, Vol. 5.
- [8] Carol A. Meyers and Andreas S. Schulz. Integer equal flows. *Oper. Res. Lett.*, 37:245–249, 2009.
- [9] David R. Morrison, Jason J. Sauppe, and Sheldon H. Jacobson. A network simplex algorithm for the equal flow problem on a generalized network. *INFORMS Journal on Computing*, 25:2–12, 2013.
- [10] C Wei, S Knust, N.V. Shakhlevich, and Stefan Waldherr. The assignment problem with nearly monge arrays and incompatible partner indices. *Discrete Applied Mathematics*, 211, 01 2015.
- [11] M. M. Zavlanos, L. Spesivtsev, and G. J. Pappas. A distributed auction algorithm for the assignment problem. In *2008 47th IEEE Conference on Decision and Control*, pages 1212–1217, Dec 2008.