in: "Advances in Difference Equations"
(Proc. 2nd Conf. on Difference Eqns and Applications. Veszprém Hungary 1995)
ed. by S. Elaydi, G. Ladas, I Gyori
Gordon and Breach 1997 p 355-362.

GENERALIZED BELLMAN EQUATION AND FLOW PROBLEMS

M.F. KONDRAT'EVA
Moscow Institute of Electronics and Mathematics,
109028, Moscow, Russia

Abstract The concept that optimization problems are linear in appropriate idempotent semirings is applied to certain flow problems. The Ford-Fulkerson and some other algorithms are represented in terms of the generalized Bellman equation. A similar approach has recently been used in some other optimization problems on graphs (see the bibliography in [1]).

1. INTRODUCTION

According to Maslov's concept [1], an idempotent linear algebra is a base of all discrete optimization algorithms on graphs. It is so because a generalized discrete Bellman equation, which describes the optimal value of the quality functional, is linear in a semimodule $\mathcal{A}^N$ over an appropriate idempotent semiring $\mathcal{A}$. Let me remind this construction.

Definition 1 A set M equipped with commutative and associative operation $\oplus$ and unit element $\mathbf{0}$: $a \oplus \mathbf{0} = a$, $\forall a \in M$, is called an *idempotent semigroup* $\mathcal{G} = (M, \oplus)$ if $a \oplus a = a$ for all $a \in M$.

Definition 2 An idempotent semigroup $\mathcal{G}$ is said to be an *idempotent semiring* $\mathcal{A} = (M, \oplus, \odot)$ if there is one more operation $\odot$ such that 1) there exists unit element $\mathbf{1}$ for $\odot$ in M; 2) $\odot$ is distributive with respect to $\oplus$; 3) $\mathbf{0} \odot a = \mathbf{0}$ for all $a \in M$.

Let $B(X, \mathcal{A})$ be a set of $\mathcal{A}$-valued functions on a discrete set $X = \{x_0, \dots, x_n\}$. It possesses a natural $\mathcal{A}$-semimodule structure w.r.t. pointwise operation $\oplus$ and pointwise (left) $\odot$-multiplication by elements of $\mathcal{A}$. Thus we have the isomorphism of $\mathcal{A}$-semimodules

$B(X,\mathcal{A}) \cong \mathcal{A}^{n+1}$. We shall consider elements of $\mathcal{A}^{n+1}$ as row-vectors. Let $H : \mathcal{A}^{n+1} \to \mathcal{A}^{n+1}$ be an endomorfism of the form

$$(H\vec{S})_j = (\oplus_{k=0}^{n} H(x_k, x_j) \odot S(x_k)), \quad \vec{S} \in \mathcal{A}^{n+1}. \tag{1.1}$$

Definition 3 The following equation for function $\vec{S} \in B(X,\mathcal{A})$ is called *generalized Bellman equation* (BE)

$$\vec{S} = H\vec{S} \oplus \vec{F}, \quad \vec{S}, \vec{F} \in \mathcal{A}^{n+1}, \tag{1.2}$$

where H is defined in (1.1). A solution of BE is calculated by formula

$$\vec{S} = (\oplus_{m=1}^{n} H^m)\vec{F}. \tag{1.3}$$

Here H^m is a composition of m endomorphisms H and, by (1.1), it is simply m-th power of matrix H with elements $H_{kj} = H(x_k, x_j) \in \mathcal{A}$.

Let me illustrate the connection between Bellman equation and optimization problems on graphs using the following simplest example.

Shortest path problem (SPP). Consider a connected, directed, arc-weighted graph $G = (X, A)$ with marked initial and final nodes. A weight $H(x_k, x_j)$ of an arc $(x_k, x_j) \in A$ means a distance from node x_k to node x_j. The problem is to find a path of minimal length from initial node x_0 to final node x_n. Here by a *path length* we mean a sum of its arcs' weights.

The algorithm for solving this problem in general case, when arc's weight may be negative, was proposed by Ford, Bellman and Moore (see [2,Ch. 8] and the references given there).

Ford-Bellman-Moore (FBM) algorithm. Let $S_m(x_j)$, $m \geq 0$, $j = 0, \ldots, n$ be a sequence of labels of node $x_j \in X$. Set

$$S_0(x_0) := 0, \quad S_0(x_j) := \infty, \, 0 < j \leq n$$

and use the iterative rule for $m \geq 0$

$$S_{m+1}(x_j) := \min\left(S_m(x_j), \min_{k=0,\ldots,n}(H(x_k, x_j) + S_m(x_k))\right), \quad 0 \leq j \leq n.$$

The sequence $S_m(x_j)$ terminates whenever there exists such $M \geq n$ that $S_{M+1}(x_j) = S_M(x_j) \equiv S(x_j)$ for all $j = 0, \ldots, n$. It means that

$$S(x_j) \leq S(x_k) + H(x_k, x_j), \quad 0 \leq j, \, k \leq n.$$

So, a value $S(x_n)$ is the required value of shortest path from x_0 to x_n.

After comparing this formulae with (1.2), (1.1) it becomes clear that the function $\vec{S}$ is a solution of BE in the semiring $(M, \oplus, \odot)$

$$M = \mathbf{R}^1 \quad \oplus \equiv \min, \quad \odot \equiv +, \quad \mathbf{0} \equiv +\infty, \quad \mathbf{1} \equiv 0 \qquad (1.4)$$

and

$$F(x_j) = \delta_{x_0}(x_j), \qquad \delta_{x_k}(x_j) = \begin{cases} \mathbf{1}, & j = k \\ \mathbf{0}, & j \neq k \end{cases}, 0 \leq j, k \leq n. \qquad (1.5)$$

Introduce a scalar product in $\mathcal{A}^{n+1}$ by formula $\langle \vec{a}, \vec{b} \rangle_{\mathcal{A}} = \oplus_{k=0}^{n} a_k \odot b_k$. Then $S(x_n) = \langle \vec{S}, \vec{\delta}_{x_n} \rangle$, where $\vec{S}$ is calculated by (1.3) for $\vec{F} = \vec{\delta}_{x_0}$.

Note that the value $(H^m)(x_k, x_j)$ in semiring (1.4) is equal to the length of shortest path of m arcs between the nodes x_k and x_j in graph G and $(H^m \vec{\delta}_{x_0})_j$ is a value of shortest m-arcs path from x_0 to x_j.

The following Generalized Wave Algorithm proposed in [3] is a generalization of FBM algorithm for case of arbitrary semiring $A = (M, \oplus, \odot)$. Here a label $S(x_j)$ of node x_j is a value of *optimal path* (e.g. shortest, critical, etc.) from x_0 to x_j and $W(x_j) = k$ if an arc (x_k, x_j) belongs to optimal path. By optimal path we mean a path of length which is optimal in sense of the partial order relation ρ connected with idempotent operation: $a \rho b$ if $a \oplus b = a$, $a, b \in M$. (In case of SPP ρ is $\leq$ and inverse order relation $\bar{\rho}$ is $>$.) The *path length* is computed by "multiplying" of its arcs' weights in sense of operation $\odot$.

Generalized Wave Algorithm(GWA).

Step 0. $\quad \vec{S}_0 := \vec{\delta}_{x_0}(1.5), \quad W_0(x_j) := \emptyset, \ 0 \leq j \leq n;$

Step $m \geq 1$.

$$\begin{aligned} &\text{If } \ \exists (x_k, x_j): \quad S_{m-1}(x_j)\, \bar{\rho}\, (S_{m-1}(x_k) \odot H(x_k, x_j)) \\ &\text{then} \quad S_m(x_j) := S_{m-1}(x_k) \odot H(x_k, x_j); \\ &\qquad\quad W_m(x_j) := k, \quad W_m(x_r) := W_{m-1}(x_r)\ \forall\, r \neq j; \end{aligned}$$

Termination condition: there exists such M that

$$S_M(x_j)\, \rho\, (S_M(x_k) \odot H(x_k, x_j)), \quad 0 \leq j, k \leq n.$$

Set $S(x_j) = S_M(x_j)$ and $W(x_j) = W_M(x_j)$, $0 \leq j \leq n$. Then $S(x_n)$ is a value of optimal path from x_0 to x_n and $(W(x_n);\ W(W(x_n));\ W(W(W(x_n))); \ldots; x_0)$ is inverse sequence of nodes of optimal path.

Thus, if an optimization problem on graph is represented in terms of BE, i.e. the corresponding semiring $\mathcal{A}$ and endomorphism H are found, then GWA can be used.

It is known that SPP has finite solution iff there are no cycles of negative length in the graph G. Rewrite this condition for the semiring

(1.4) in the form

$$H^m(x_j,x_j)\oplus \mathbf{1}=\mathbf{1},\quad 0\le j\le n,\quad m=1,2,\dots,n+1. \qquad (1.6)$$

In case of general semiring condition (1.6) is sufficient for existence of a finite solution of BE and finiteness of steps number in GWA.

The aim of my work is to consider known algorithms for the maximum flow problem (MFP) and minimum cost flow problem (MCFP) from the point of view of BE and GWA. The possibility to do this was shortly discussed in [6]. The main idea is that at each iteration of multiiterative procedure for finding optimal flow we have to solve BE in certain semiring. Such semirings are found and a condition for flow to be optimal is formulated in terms of a solution of BE.

2. THE MAXIMUM FLOW PROBLEM

Given a connected, directed graph $G=(X,A)$, associate with each arc $(x_j,x_k)\in A$ a positive capacity $Q(x_j,x_k)$. Besides that, we have two special nodes: sourse x_0 and sink x_n, i.e. there are no arcs in A entering x_0 and outgoing from x_n. Such graph G is called a *network.*

Definition A nonnegative function $\Phi(x_j,x_k)$ running over all arcs $(x_j,x_k)\in A$ is called a *flow* in a network G if

$$1) 0\le\Phi(x_j,x_k)\le Q(x_j,x_k),$$
$$2)\sum_{k=0}^{n}\Phi(x_j,x_k)-\sum_{r=0}^{n}\Phi(x_r,x_j)=\begin{cases} v, & j=0\\ -v, & j=n,\\ 0, & \text{otherwise}\end{cases} \qquad (2.1)$$

and the value $v(\Phi)\equiv v$ is said to be the *value of flow.*
The problem is to find a flow in a network with the maximum value.

The Ford-Fulkerson (FF) algorithm [4, Ch. 1] starts with an arbitrary flow Φ_0 in a network G and constructs a sequence Φ_m of flows in G until the maximum flow is obtained. Given a graph G and a flow Φ, consider an *incremental* graph G_Φ with the same nodes as G. The ordered pair (x_j,x_k) is an arc of G_Φ iff either $(x_j,x_k)\in A$ and $H(x_j,x_k)\equiv Q(x_j,x_k)-\Phi(x_j,x_k)>0$, or $(x_k,x_j)\in A$ and $H(x_j,x_k)\equiv\Phi(x_k,x_j)>0$. The subsets of arcs of the first and the second type we denote respectively by A_+ and A_-. Call $H(x_j,x_k)$ the weigth of arc (x_j,x_k) in G_Φ. Let $\pi(x,y)$ be a path in G_Φ from $x\in X$ to $y\in X$. We call the minimum value of its arcs' weights the *path length*: $\epsilon(x,y) = \min_{(x_j,x_k)\in\pi(x,y)}(H(x_j,x_k))$. The FF labeling method requires a process for finding an *augmenting* path π_* in G_{Φ_m} i.e. a path $\pi(x_0,x_n)$ of positive length $\epsilon(x_0,x_n) > 0$. Changing the flow on arcs of an augmenting path, we increase the value of flow by $\epsilon = \epsilon(x_0,x_n)$. We write $\Phi_{m+1} = \Phi_m + \epsilon\cdot\pi_*$ which means

$$\Phi_{m+1}(x_j,x_k) = \begin{cases} \Phi_m(x_j,x_k)+\epsilon, & (x_j,x_k)\in A_+\bigcap\pi_*, \\ \Phi_m(x_j,x_k)-\epsilon, & (x_j,x_k)\in A_-\bigcap\pi_*, \\ \Phi_m(x_j,x_k), & (x_j,x_k)\notin\pi_* \end{cases} \tag{2.2}$$

for $0\le j,k\le n$. A flow Φ in G is maximum iff $\epsilon(x_0,x_n)=0$ for all paths $\pi(x_0,x_n)$ in incremental graph G_Φ.

For integer capacities $Q(x_i,x_j)$, the sequence of flows Φ_m terminates at $m\le v_{\max}$. The following rule proposed in [5] provides better estimation for number of augmentations $m\le 1+\log v_{\max}$:

(i) *each augmentation is done along an augmenting path giving the maximum possible augmentation.*

If capacities Q are incommensurable, sequence Φ_m may either not terminate or converge to a nonmaximum flow [4]. To avoid this, we use the rule that guarantees $m\le(n^3-n)/4$ for a n-node network [5]:

(ii) *each augmentation is done along a path giving the maximum augmentation among paths having fewest arcs.*

Consider rule (i). So, we need a path with maximum length $\epsilon(x_0,x_n)$. Assign to this problem the semiring $\mathcal{A}=(M,\oplus,\odot)$ with

$$M=\mathbf{R}_+,\quad \oplus\equiv\max,\quad \odot\equiv\min,\quad \mathbf{0}\equiv 0,\quad \mathbf{1}\equiv\infty. \tag{2.3}$$

For the endomorphism H and the function $\vec{F}$ of the form

$$H(x_j,x_k)=(Q(x_j,x_k)-\Phi(x_j,x_k))\oplus\Phi(x_k,x_j),\quad F(x_j)=\delta_{x_0}(x_j) \tag{2.4}$$

a solution of BE gives the required value $\epsilon(x_0,x_n)=S(x_n)$.

Consider rule (ii). Now a *path length* is a 2-vector $\vec{\epsilon}$. Its components ϵ_1 and ϵ_2 are the number of arcs and the scalar length ϵ. We need to minimize ϵ_1 and maximize ϵ_2. Corresponding semiring is

$$M=\{\vec{a}=(a_1,a_2): a_1=\begin{cases} a_1, & a_2\ne 0, \\ \infty, & a_2=0, \end{cases}\ a_1,a_2\in\mathbf{R}_+\}, \tag{2.5}$$

$$\begin{gathered}\vec{a}\oplus\vec{b}=\begin{cases} \vec{a}, & \text{if}\ (a_1<b_1)\ \text{or}\ (a_1=b_1\ \text{and}\ a_2\ge b_2), \\ \vec{b}, & \text{if}\ (b_1<a_1)\ \text{or}\ (a_1=b_1\ \text{and}\ b_2\ge a_2), \end{cases} \\ \vec{a}\odot\vec{b}=(a_1+b_1,\min(a_2,b_2)),\quad \mathbf{0}=(\infty,0),\quad \mathbf{1}=(0,\infty),\end{gathered} \tag{2.6}$$

and the endomorphism and inhomogeneity in BE are

$$H(x_j, x_k) = \vec{h}^{(jk)} = (h_1^{(jk)}, h_2^{(jk)}), \quad \vec{F} = \vec{\delta}_{x_0},$$
$$h_2^{(jk)} = H(x_j, x_k) \text{ from (2.4)}, \; h_1^{(jk)} = \begin{cases} 1, & h_2^{(jk)} \neq 0, \\ \infty, & h_2^{(jk)} = 0. \end{cases} \tag{2.7}$$

A solution of BE in this case is a vector $\vec{S}$ with 2-vector components $S_j = \vec{s}^{(j)} \equiv (s_1^{(j)}, s_2^{(j)})$, $0 \leq j \leq n$. Here $s_1^{(j)}$ is the minimal number of arcs that form a path from x_0 to x_j in G_Φ and $s_2^{(j)}$ is the maximal length $\epsilon(x_0, x_j)$ of those paths. So, $\epsilon(x_0, x_n) = s_2^{(n)}$. Finally, we obtain

Theorem 1 *The process of finding an augmenting path by the rules (i) or (ii) and augmentation value for a flow in FF algorithm is equivalent to solving BE (1.2) with (2.3), (2.4) or (2.5),(2.6),(2.7) respectively. Condition*

$$\langle \vec{S}, \vec{\delta}_{x_n} \rangle_{\mathcal{A}} = \mathbf{0} \tag{2.8}$$

in corresponding semiring is the criterion for a flow to be maximum.

3. THE MINIMUM-COST FLOW PROBLEM

Given a network $G = (X, A)$, associate with each arc (x_j, x_k) a non-negative cost $C(x_j, x_k)$ as well as the positive capacity $Q(x_j, x_k)$. Set by definition $C(x_j, x_k) = \infty$ and $Q(x_j, x_k) = 0$ iff $(x_j, x_k) \notin A$. Let the cost of a flow Φ be $c(\Phi) = \sum_{(x,y)\in A} \Phi(x, y) \cdot C(x, y)$ and its value be $v(\Phi)$ (2.1). Call a flow Φ *extreme* if it is of minimum cost among flows with the value $v = v(\Phi)$. The problem is to find an extreme maximum flow.

Assume for convenience that $(x_j, x_k) \in A \Rightarrow (x_k, x_j) \notin A$.

To solve the problem we need an incremental graph G_Φ with the same structure as in Section 2 and the vector-valued arcs' weights

$$H(x_j, x_k) = \vec{h}^{(jk)} = \begin{cases} \begin{pmatrix} C(x_j, x_k) \\ Q(x_j, x_k) - \Phi(x_j, x_k) \end{pmatrix}, & (x_j, x_k) \in A_+, \\ \begin{pmatrix} -C(x_k, x_j) \\ \Phi(x_k, x_j) \end{pmatrix}, & (x_j, x_k) \in A_-. \end{cases} \tag{3.1}$$

Set $h_1 = \infty$ for all $\vec{h}$ with $h_2 = 0$.

Thus for MCFP the following set M arises

$$M = \{\vec{a} = (a_1, a_2) : a_1 = \begin{cases} a_1, & a_2 \neq 0, \\ \infty, & a_2 = 0, \end{cases} \; a_1 \in \mathbf{R}, \, a_2 \in \mathbf{R}_+\}. \tag{3.2}$$

A *path length* in graph G_Φ is 2-vector $\vec{\epsilon} = (\epsilon_1, \epsilon_2)$ which is $\odot$-product of arcs' weights (3.1) of the path w.r.t. operation $\odot$ from (2.6). Call the value ϵ_1 a *path cost.* A directed cycle in G_Φ is said to be *negative* if it has a negative cost. According to known theorem [2, Ch. 11], Φ is extreme in G iff there are no negative cycles in G_Φ. In BE formalism this condition is exactly (1.6) for function H (3.1) and operations (2.6).

The algorithm for constructing an extreme flow, based on finding negative cycles in G_Φ, is following [2, Ch. 11].

1. Given Φ_m, $m \geq 0$ in G with value $v(\Phi_m)$, construct G_{Φ_m}.
2. If there are no negative cycles in G_{Φ_m} then stop; Φ_m is extreme.
3. If ν is a negative cycle in G_{Φ_m} with length $\vec{\epsilon}^{\,\nu} = (\epsilon_1^\nu, \epsilon_2^\nu)$ then add to Φ_m an augmenting flow Φ^ν changing Φ_m only on arcs of the cycle ν: $\Phi_{m+1} = \Phi_m + \Phi^\nu \equiv \Phi_m + \epsilon_2^\nu \cdot \nu$ (here notation (2.2) is used). Then we obtain a new flow Φ_{m+1} with the same value $v(\Phi_{m+1}) = v(\Phi_m)$, but lower cost : $c(\Phi_{m+1}) = c(\Phi_m) + c(\Phi^\nu)$, where $c(\Phi^\nu) = \epsilon_1^\nu \cdot \epsilon_2^\nu < 0$.

Statement 1 *Let H be a matrix with vector-valued elements $H_{jk} = H(x_j, x_k)$, defined in (3.1) and H^m be an m-th power of H in the semiring (3.2),(2.6). If there exist $m_0 \leq n+1$ and $x \in X$ such that $H^m(x,x) \oplus \mathbf{1} = \mathbf{1}$ for $m < m_0$, and $H^{m_0}(x,x) \oplus \mathbf{1} = H^{m_0}(x,x)$ then there exists an m_0-arc negative cycle ν in the network G_Φ such that x is one of its nodes and $\vec{\epsilon}^{\,\nu} = H^{m_0}(x,x)$.*

The algorithm above starts with a flow of fixed value and decreases its cost while possible. Another way is to start with the zero flow and to increase its value, keeping at every step a property of extremality of a flow [4,Ch. 3], [2,Ch. 11]. This algorithm is based on finding a path of minimal cost in G_Φ and the following theorem: *if Φ_m is an extreme flow in G with value $v(\Phi_m) = v$ and π is a path of minimal cost from x_0 to x_n in G_{Φ_m} with lenght $\vec{\epsilon}^{\,\pi} = (\epsilon_1^\pi, \epsilon_2^\pi)$ then $\Phi_{m+1} = \Phi_m + \epsilon_2^\pi \cdot \pi$ is an extreme flow with value $v(\Phi_{m+1}) = v + \epsilon_2^\pi$.* So we need to find the path π and to add an augmenting flow Φ^π that changes a flow Φ_m on arcs of the path π and has value $v(\Phi^\pi) = \epsilon_2^\pi$. The cost of the flow Φ_{m+1} is $c(\Phi_{m+1}) = c(\Phi_m) + c(\Phi_\pi)$, where $c(\Phi^\pi) = \epsilon_1^\pi \cdot \epsilon_2^\pi$ is the cost of augmenting flow Φ^π. The following statement gives us an expression for the vector $\vec{\epsilon}^{\,\pi}$ through a solution of BE.

Statement 2 *If $\vec{S}$ is a solution of BE in semiring (2.6), (3.2) with endomorphism H (3.1) and $\vec{F} = \vec{\delta}_{x_0}$ then $\vec{\epsilon}^{\,\pi} = S(x_n) \equiv \vec{s}^{(n)}$ and the path π is obtained as a result of GWA in this semiring.*

The criterion for extreme flow to be maximum is (2.8) in semiring (2.6), (3.2) and $\vec{S}(x)$ from Statement 3. Indeed, it means that $s_2^{(n)} = 0$ i.e. we can't increase the value of flow.

4. CONCLUSION

A flow problem can be reduced to a path problem i.e. every iteration in multiiterative procedure for finding the optimal flow is a procedure of seeking an augmenting path with certain property. This property defines the originality of an algorithm. On the other hand, generalized Bellman equation describes an optimal path, and all specific features are represented in realization of a semiring $\mathcal{A} = (M, \oplus, \odot)$ and endomorphism H. An operation $\odot$ defines a way of path length calculation, and an operation $\oplus$ defines an order on path lengths. So, it is natural that flow problems admit BE formalization. I'd like to underline that fixing a semiring and BE, you fix a problem togeter with an idea of its solving.

ACKNOLEDGEMENTS

I am greatly indebted to S.M. Avdoshin and V.V. Belov for suggesting the problem and for drawing my attention to this topic of researches.

REFERENCES

[1] V.N. Kolokoltsov, V.P. Maslov, *Idempotent Analysis and Its Applications to Optimal Control Theory*, Nauka, Moscow, 1994 (in Russian).

[2] N. Christofides, *Graph Theory. An Algorithmic Approach*, Academic Press, New York-London-San Francisco, 1975.

[3] S.M. Avdoshin, V.V. Belov, Generalized Wave Algorithm for Solving Extremal Problems on Graphs, *J. Vychislit. Math. i Math. Phys.*, **19:3** (1979), 739–755.

[4] L.R. Ford, D.R. Fulkerson, *Flows in Networks*, Princeton Univ. Press, 1962.

[5] J. Edmonds, R.M. Karp, Theoretical Improvements in Algorithmic Efficiency for Network Flow Problems, *J. ACM* **19:2** (1972), 248–264.

[6] S.M. Avdoshin, V.V. Belov, V.P. Maslov, V.M. Piterkin, *Optimization of Flexible Industrial Systems*, MIEM, Moscow, 1987 (in Russian).