The auction algorithm

Giovanni Righini

University of Milan

UNIVERSITÀ DEGLI STUDI DI MILANO

The auction algorithm (Bertsekas, 1979)

The auction algorithm was conceived for solving the linear assignment problem (i.e. finding an optimal matching in a bipartite weighted graph) through parallel computation.

It can be interpreted as a primal-dual algorithm, in which both primal and dual solutions can worsen in some iteration.

Its functioning resembles a sequential distributed decision process in which n persons bid for n objects, as in an auction-based market.

Its convergence is based on ϵ-complementary slackness: CSCs violations are allowed within a given amount.

Primal problem

Data.

- a set P of n persons and a set O of n objects;
- a revenue r_{ij} of each object $j \in O$ for each person $i \in P$.

Variables: $x_{ij} \in \{0, 1\}$: assignment of object j to person i.
Integrality constraint can be relaxed.

Revenue maximization problem:

$$
\begin{aligned}
\text{P) maximize } z = \sum_i \sum_j r_{ij} x_{ij} & & \\
\text{s.t. } \sum_j x_{ij} = 1 & \quad \forall i \in P & [p_i] \\
\sum_i x_{ij} = 1 & \quad \forall j \in O & [c_j] \\
x_{ij} \geq 0 & \quad \forall i, j &
\end{aligned}
$$

Economic interpretation

Assume that each object $j \in O$ can be assigned a cost c_j by the persons.

The net profit for each person $i \in P$ assigned to object $j \in O$ is $r_{ij} - c_j$. For any given feasible assignment x,

$$p_i = \sum_{j \in O} (r_{ij} - c_j) x_{ij}.$$

For a given vector of costs c, the maximum profit obtainable by each person i is

$$p_i^* = \max_{j \in O} \{r_{ij} - c_j\}.$$

Equilibrium

For a given assignment x and a given vector of costs c, a person i is happy iff he is assigned an object j such that his profit is maximum:

$$\sum_{j \in O} (r_{ij} - c_j) x_{ij} = \max_{j \in O} \{ r_{ij} - c_j \}.$$

The costs c (dual variables) and the assignments x (primal variables) are at equilibrium iff all persons are happy.

The dual problem

$$\text{D) minimize } w = \sum_i p_i + \sum_j c_j$$

$$\text{s.t. } p_i + c_j \geq r_{ij} \quad \forall i, j \quad [x_{ij}]$$

$$p_i \text{ unrestricted} \quad \forall i$$

$$c_j \text{ unrestricted} \quad \forall j.$$

Since

$$p_i^* = \max_{j \in O}\{r_{ij} - c_j\},$$

the optimal dual value is

$$w^* = \sum_i p_i^* + \sum_j c_j^* = \sum_j c_j^* + \sum_i \max_{j \in O}\{r_{ij} - c_j^*\}$$

and, in general,

$$w = \sum_j c_j + \sum_i \max_{j \in O}\{r_{ij} - c_j\}.$$

Complementary slackness

Primal complementary slackness conditions:

$$x_{ij}(p_i + c_j - r_{ij}) = 0 \;\; \forall i \in P, j \in O.$$

When object j is assigned to person i

$$p_i = r_{ij} - c_j \;\; \forall (i,j) : x_{ij} = 1.$$

Otherwise,

$$p_i \geq r_{ij} - c_j \;\; \forall (i,j) : x_{ij} = 0$$

is the optimality condition: any alternative assignment would not be an improvement for any person.

Weak duality

For any feasible assignment, $\sum_i x_{ij} = 1 \ \forall j$. Hence

$$c_j = \sum_i c_j x_{ij}.$$

Using this substitution:

$$\begin{aligned} w(c) &= \sum_j c_j + \sum_i \max_j \{r_{ij} - c_j\} \geq \\ &\geq \sum_i \sum_j c_j x_{ij} + \sum_i \sum_j (r_{ij} - c_j) x_{ij} = \\ &= \sum_i \sum_j r_{ij} x_{ij} = z(x). \end{aligned}$$

Hence

$$w(c^*) \geq z(x^*).$$

The algorithm

At each iteration, if equilibrium has not been achieved, then at least one person $i \in P$ is not happy.

Select the best and the second best choice for i:

$$k' = \arg\max_{j \in O}\{r_{ij} - c_j\}$$

$$k'' = \arg\max^2_{j \in O}\{r_{ij} - c_j\}$$

Let p be the person currently assigned to k'.

- the assignments to persons i and p are swapped;
- the cost of object k' is raised to the value at which person i is indifferent between k' and k''.

Degenerate iterations

The algorithm may cycle forever in some cases, because of denegerate iterations in which the cost increase is

$$\max_{j \in O}\{r_{ij} - c_j\} - \overset{2}{\max_{j \in O}}\{r_{ij} - c_j\} = 0.$$

To overcome this problem, Bertsekas proposed to resort to ϵ-optimality.

ϵ-optimality

A solution (x, c) is ϵ-optimal if and only if

$$\sum_j (r_{ij} - c_j)x_{ij} \geq \max_j \{r_{ij} - c_j\} - \epsilon \tag{1}$$

i.e. each person is almost happy. Any ϵ-optimal solution satisfies the CSC when they are relaxed into

$$x_{ij}(p_i + c_j - r_{ij}) \leq \epsilon \;\; \forall i \in P, j \in O.$$

Proof. For all (i, j) such that $x_{ij} = 0$, the inequality is trivially satisfied, since $\epsilon \geq 0$. For all (i, k) such that $x_{ik} = 1$, it holds:

- $\sum_j (r_{ij} - c_j)x_{ij} = r_{ik} - c_k$ (feasible assignment);
- $p_i \leq p_i^* = \max_j\{r_{ij} - c_j\}$ (by definition of p_i^*);

Hence (1) implies

$$r_{ik} - c_k \geq p_i - \epsilon$$

$$p_i + c_k - r_{ik} \leq \epsilon \;\; \square$$

When $\epsilon = 0$, $z^* = w^*$ (strong duality for linear programming).

ϵ-optimality

If all persons are almost happy (within $\epsilon \geq 0$),

$$\sum_j (r_{ij} - c_j) x_{ij} \geq \max_j \{r_{ij} - c_j\} - \epsilon \quad \forall i \in P. \tag{2}$$

Summing up for all $i \in P$,

$$\sum_i \sum_j (r_{ij} - c_j) x_{ij} \geq \sum_i \max_j \{r_{ij} - c_j\} - n\epsilon$$

$$\sum_i \sum_j r_{ij} x_{ij} \geq \sum_i \max_j \{r_{ij} - c_j\} + \sum_i \sum_j c_j x_{ij} - n\epsilon$$

Replacing $\sum_{i \in P} \sum_{j \in O} c_j x_{ij}$ with $\sum_{j \in O} c_j$,

$$\sum_i \sum_j r_{ij} x_{ij} \geq \sum_i \max_j \{r_{ij} - c_j\} + \sum_{j \in O} c_j - n\epsilon.$$

Primal-dual interpretation

$$\sum_i \sum_j r_{ij} x_{ij} \geq \sum_i \max_j \{r_{ij} - c_j\} + \sum_{j \in O} c_j - n\epsilon.$$

From the definitions

$$z(x) = \sum_{i \in P} \sum_{j \in O} r_{ij} x_{ij}$$

$$w(p) = \sum_{i \in P} \max_{j \in O} \{r_{ij} - c_j\} + \sum_{j \in O} c_j,$$

it follows

$$z(x) \geq w(p) - n\epsilon.$$

Since $z^* = w^* \leq w(p)$,

$$z(x) \geq z^* - n\epsilon,$$

i.e. the primal solution is $n\epsilon$-optimal.

The modified algorithm

At each iteration, when person i bids for object k', being k'' the second best choice, the cost increase of k' is set to

$$\max_{j \in O}\{r_{ij} - c_j\} - \overset{2}{\max_{j \in O}}\{r_{ij} - c_j\} + \epsilon,$$

so that it strictly positive and infinite loops are prevented.

Termination

Just after bidding for an object k, a person i is almost happy.

It remains almost happy while he keeps holding object k, because the costs of the other objects monotonically increase and therefore none of them can become more attractive than object k for person i.

Therefore, persons that are not almost happy must be assigned to objects that have not received any bid.

Therefore, if all objects have received at least one bid, all persons must be almost happy and the algorithm stops.

Any object that receives m bids, has a cost increase of $m\epsilon$. After a large enough (but finite) number of iterations, no object can remain without any bid.

Hence, the algorithm terminates in a finite number of steps.

Scaling

When the algorithm stops, the solution is ϵ-optimal.

If data are integer and $\epsilon < 1/n$, then ϵ-optimality implies optimality.

Alternatively, scaling phases can be done with decreasing values of ϵ, using the final assignment x as the initial assigment for the next one.

No worst-case bound is provided to the number of phases, but experimental results show very short running times.