Epsilon-Constraint Implementation in Online Optimization Problems

D. Gueorguiev 2/22/24

Table of Contents

The Epsilon Constraint Technique

The epsilon constraint method can properly generate the Pareto optimal solutions of Multi-objective Optimization problems and find the corresponding efficient set. In the new Optimization Model the epsilon constraint method is used to find the efficient fulfillment options in tandem with the Split Recommender Algorithm. The latest version of the Split Recommender Algorithm (denoted as SRA v4) has built-in dominance enforcing decision code which guarantees that non-dominated combination of delay, shipping cost, and CO2 emissions will be selected as the final Fulfillment Decision where the Delay will have precedence with respect to the other two objectives. For details on SRA v4 and corresponding pseudocode see [this document].

A Bit of Multi-Objective Optimization Theory

Let us define the multi-objective optimization problem

$$\min\left(f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_p(\mathbf{x})\right)$$
$$\text{subject to } \mathbf{x} \in \mathcal{X}$$

for which we want to find the set of feasible solutions which would minimize the set of objectives $f_i, i = 1..p$ over the set $\mathcal{X}$. We denote with $\mathcal{X}$ is the *decision space* of the multi-objective optimization problem. $\mathbf{x} \in \mathcal{X}$ will denote feasible solution of the multi-objective optimization problem.

For Fulfillment optimization the decision space is defined by the tuple $(\mathcal{O}, \mathcal{E})$ where $\mathcal{O}$ denotes the order space and $\mathcal{E}$ denotes the space of the environment state. Both spaces are represented by a set of attributes which in turn are sets of quantities with various domains (discrete and continuous). For instance $\mathcal{O}$ consists of the set of GTINs in the order, a map between GTINs in the order and the requested quantities for each. The environment $\mathcal{E}$ consists of node inventories, node capacities, dual variables, objective values (aka rewards) accumulated so far, etc. Each objective function f_i maps the current order $\sigma \in \mathcal{O}$ and the current environment state $e \in \mathcal{E}$ to a real number which has cost semantics where f_1 represents delay, f_2 represents shipping cost and f_3 represents CO2 emissions. Thus, f in Fulfillment optimization is a map from $\mathcal{O} \times \mathcal{E}$ to $\mathbb{R}^3$.

The space $\mathcal{Y} \equiv \mathbb{R}^p$ is the *objective* (aka *criterion*) *space* of the problem.

Definition: *componentwise order* $\leq$
The relation $\leq$ defined in p-dimensional space (e.g. $\mathbb{R}^p$) denotes the *componentwise order* where $\mathbf{y}^{(1)} \leq \mathbf{y}^{(2)}$ ($\mathbf{y}^{(1)}, \mathbf{y}^{(2)} \in \mathbb{R}^p$) if and only if $y_k^{(1)} \leq y_k^{(2)}, k = 1, \dots, p$ and $\mathbf{y}^{(1)} \neq \mathbf{y}^{(2)}$.

Definition: weak componentwise order $\leqq$
The relation $\leqq$ in p-dimensional space (e.g. $\mathbb{R}^p$) denotes the weak *componentwise order* where $\mathbf{y}^{(1)} \leq \mathbf{y}^{(2)}$ ($\mathbf{y}^{(1)}, \mathbf{y}^{(2)} \in \mathbb{R}^p$) if and only if $y_k^{(1)} \leq y_k^{(2)}, k = 1, \dots, p$.

Notation: the point in objective space represented by the tuple $\left(f_1(\mathbf{x}), f_2(\mathbf{x}), f_3(\mathbf{x}), \dots, f_p(\mathbf{x})\right)$ will be simply denoted as $f(\mathbf{x})$ where $f : \mathcal{X} \to \mathcal{Y}$.

Definition: *non-negative orthant* of $\mathbb{R}^p$, the *non-negative orthant* of $\mathbb{R}^p$ *without the origin* and the *positive orthant* of $\mathbb{R}^p$
The *non-negative orthant* of $\mathbb{R}^p$ is defined as $\mathbb{R}^p_{\geqq} \coloneqq \{\mathbf{y} \in \mathbb{R}^p : \mathbf{y} \geqq 0\}$
The *non-negative orthant* of $\mathbb{R}^p$ *without the origin* is defined as $\mathbb{R}^p_{\geq} \coloneqq \{\mathbf{y} \in \mathbb{R}^p : \mathbf{y} \geq 0\} = \mathbb{R}^p_{\geqq} \setminus \{0\}$
The positive orthant of $\mathbb{R}^p$ is defined as $\mathbb{R}^p_{>} \coloneqq \{\mathbf{y} \in \mathbb{R}^p : \mathbf{y} > 0\} = \operatorname{int} \mathbb{R}^p_{\geqq}$
Note: for $p = 1$ obviously $\mathbb{R}_{\geq} \equiv \mathbb{R}_{>}$.

Definition *efficient* (aka *Pareto-optimal*) *solution* and *efficient set*
A feasible solution $\hat{\mathbf{x}} \in \mathcal{X}$ is called *efficient* or *Pareto optimal* if there is no other $\mathbf{x} \in \mathcal{X}$ such that $f(\mathbf{x}) \equiv \left(f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_p(\mathbf{x})\right) \leq f(\hat{\mathbf{x}}) \equiv \left(f_1(\hat{\mathbf{x}}), f_2(\hat{\mathbf{x}}), \dots, f_p(\hat{\mathbf{x}})\right)$. Note: the relation $\leq$ denotes componentwise order as defined earlier. If $\mathbf{x}$ is efficient, $f(\mathbf{x}) \equiv \left(f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_p(\mathbf{x})\right)$ is called *non-dominated point* in objective space. If $\mathbf{x}^{(1)}, \mathbf{x}^{(2)} \in \mathcal{X}$ and $f\left(\mathbf{x}^{(1)}\right) \equiv \left(f_1\left(\mathbf{x}^{(1)}\right), f_2\left(\mathbf{x}^{(1)}\right), \dots, f_p\left(\mathbf{x}^{(1)}\right)\right) \leq f\left(\mathbf{x}^{(2)}\right) \equiv \left(f_1\left(\mathbf{x}^{(2)}\right), f_2\left(\mathbf{x}^{(2)}\right), \dots, f_p\left(\mathbf{x}^{(2)}\right)\right)$ we say that $\mathbf{x}^{(1)}$ dominates $\mathbf{x}^{(2)}$ and $f\left(\mathbf{x}^{(1)}\right)$ dominates $f\left(\mathbf{x}^{(2)}\right)$.
The set of all efficient solutions $\hat{\mathbf{x}} \in \mathcal{X}$ is denoted with $\mathcal{X}_E$ and is called *efficient set*.
The set of all nondominated points $\hat{\mathbf{y}} = f(\hat{\mathbf{x}}) \in \mathcal{Y}$, where $\hat{\mathbf{x}} \in \mathcal{X}_E$ will be denoted with $\mathcal{Y}_N$ and will be called the *nondominated set*.

Let us assume that $\mathcal{X}_E$ and $\mathcal{Y}_N$ are non-empty, and want to find real numbers $\underline{y}_k, \overline{y}_k, k = 1 \dots p$ with $\underline{y}_k \leq y_i \leq \overline{y}_k$ for all $\mathbf{y} \in \mathcal{Y}_N$.

<u>Question</u>: How do we estimate the points $\underline{y}_k, \overline{y}_k$?
An obvious possibility would be to choose $\underline{y}_k \coloneqq \min_{y \in \mathcal{Y}} y_i$ **(i)** and $\overline{y}_k = \max_{y \in \mathcal{Y}} y_i$ **(ii)**. While the lower bound **(i)** is tight (there is always efficient point $\mathbf{y} \in \mathcal{Y}_N$ with $y_k = \underline{y}_k$) the upper bound **(ii)** tends to be far away from the non-dominated points. For this reason, the upper bound is defined as the maximum over nondominated points only. Thus we have:

Definition: *ideal point* and *nadir point*
The point $\mathbf{y}^I = \left(y_1^I, \dots, y_p^I\right)$ given by

$$y_k^I \coloneqq \min_{x \in \mathcal{X}} f_k(x) = \min_{y \in \mathcal{Y}} y_k \quad (1)$$

is called *the ideal point* of the multi-objective optimization problem $\min_{x \in \mathcal{X}} \left(f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_p(\mathbf{x})\right)$
The point $\mathbf{y}^N \in \left(y_1^N, \dots, y_p^N\right)$ given by:

$$y_k^N \coloneqq \max_{x \in \mathcal{X}_E} f_k(x) = \max_{y \in \mathcal{Y}_N} y_k \quad (2)$$

is called *the nadir point* of the multi-objective optimization problem.
Obviously we have $y_k^I \leq y_k$ and $y_k \leq y_k^N$ for any $\mathbf{y} \in \mathcal{Y}_N$. Furthermore, $\mathbf{y}^I$ and $\mathbf{y}^N$ are tight lower and upper bounds on the efficient set. Since the ideal point is found by solving p single objective optimization problems its computation can be considered easy. On the other hand , computation of $\mathbf{y}^N$ involves optimization over the efficient set which is **a difficult problem**.

Due to difficulty of computing $\mathbf{y}^N$ heuristics are often used. A basic estimation of the nadir point uses payoff tables. We describe one approach which attempts to estimate the nadir point below:

Definition: *payoff table*
We consider p single objective optimization problems $\min_{x \in \mathcal{X}} f_k(x), k = 1 \dots p$. Let the optimal solutions be $x^k, k = 1, \dots, p$, i.e. $f_k(x^k) = \min_{x \in \mathcal{X}} f_k(x)$. Using these optimal solutions which can be computed easily we obtain the payoff table as the Cartesian product of the two sets $\{x^k\} \times \{f_l(x^k)\}$, $k, l = 1, \dots, p$. In tabular form this Cartesian product is visualized as shown below:

	x^1	x^2	$\cdots$	x^{p-1}	x^p
f_1	y_1^I	$f_1(x^2)$	$\cdots$	$f_1(x^{p-1})$	$f_1(x^p)$
f_2	$f_2(x^1)$	$\ddots$	$\cdots$	$\cdots$	
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
f_{p-1}	$f_{p-1}(x^1)$	$\cdots$	$\cdots$	$\ddots$	$f_{p-1}(x^p)$
f_p	$f_p(x^1)$	$f_p(x^2)$	$\cdots$	$f_p(x^{p-1})$	y_p^I

Table: payoff table for multi-objective optimization problem defined over objective space $\mathcal{Y} \equiv \mathbb{R}^p$

From the payoff table it is clear that $y_k^I = f_k(x^k), k = 1 \dots p$. We define

$\tilde{y}_i^N := \max_{k=1,..,p} f_i(x^k)$ (3)

which is the largest element of row i in the payoff table as an estimate for y_i^N.

Question: is $\tilde{\mathbf{y}}^N$ obtained via (3) a good estimate of $\mathbf{y}^N$?

Answer: In general the answer is **No**. $\tilde{\mathbf{y}}^N$ may over-estimate or underestimate the nadir point $\mathbf{y}^N$ **with arbitrary error** in case $p > 2$. As mentioned earlier estimating the nadir for $p > 2$ is a difficult problem and there no efficient algorithms for doing that known to date.

Definition: *weakly and strictly efficient solutions*
A feasible solution $\hat{\mathbf{x}} \in \mathcal{X}$ is called *weakly efficient* (also *weakly Pareto optimal*) if there is no $\mathbf{x} \in \mathcal{X}$ such that $f(\mathbf{x}) < f(\hat{\mathbf{x}})$ i.e. $f_k(\mathbf{x}) < f_k(\hat{\mathbf{x}}), k = 1 \dots p$. The point $\hat{\mathbf{y}} = f(\hat{\mathbf{x}})$ is then called *weakly non-dominated*.
A feasible solution $\hat{\mathbf{x}} \in \mathcal{X}$ is called *strictly efficient* (also *strictly Pareto optimal*) if there is no $\mathbf{x} \in \mathcal{X}, \mathbf{x} \neq \hat{\mathbf{x}}$ such that $f(\mathbf{x}) \leqq f(\hat{\mathbf{x}})$. The weakly (strictly) efficient and nondominated sets are denoted $\mathcal{X}_{wE}$ $(\mathcal{X}_{sE})$ and $\mathcal{Y}_{wE}$. Note: the set $\mathcal{Y}_{sE}$ is **not defined** as there is no such concept as strict non-dominance for sets $\mathcal{Y} \subset \mathbb{R}^p$.

From the last definition we notice that a weakly nondominated point is a nondominated point with respect to the interior of *the non-negative orthant* of $\mathbb{R}^p$: $\text{int}\, \mathbb{R}^p_{\geqq} \equiv \mathbb{R}^p_{>}$.
Also from the last definition it follows that $\mathcal{Y}_N \subset \mathcal{Y}_{wN}$ and $\mathcal{X}_{sE} \subset \mathcal{X}_E \subset \mathcal{X}_{wE}$.
Equivalent definitions of weak efficiency
A feasible solution $\hat{\mathbf{x}} \in \mathcal{X}$ is weakly efficient if and only if:
1) there does not exist $\mathbf{x} \in \mathcal{X}$ such that $f(\hat{\mathbf{x}}) - f(\mathbf{x}) \in \text{int}\, \mathbb{R}^p_{\geqq} \equiv \mathbb{R}^p_{>}$
2) $\left(f(\hat{\mathbf{x}}) - \mathbb{R}^p_{>}\right) \cap \mathcal{Y} = \emptyset$

The epsilon-constraint method

Assume the following Multi-objective optimization problem:

$$\min\left(f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_p(\mathbf{x})\right) \qquad \textbf{(I)}$$

subject to : $\mathbf{x} \in \mathcal{X}$

In the epsilon-constraint method we optimize one of the objective functions using the other objective functions as constraints incorporating them in the constraint part of the model:

$$\min f_1(\mathbf{x}) \qquad \textbf{(II)}$$

subject to:

$$f_2(\mathbf{x}) \geq e_2$$
$$f_3(\mathbf{x}) \geq e_3$$
$$\ldots$$
$$f_p(\mathbf{x}) \geq e_p$$
$$\mathbf{x} \in \mathcal{X}$$

By parametrical variation of the constrained objective functions e_i the efficient solutions of the problem are obtained.

Proposition: The optimal solution of (II) is guaranteed to be efficient only if all the $(p-1)$ objective functions' constraints are binding.
Otherwise, if there are alternative optima that may improve one of the non-binding constraints that correspond to an objective function, the obtained optimal solution of the problem (II) is not in fact efficient, but it is *a weakly efficient solution*.

In order to overcome this deficiency of the original epsilon constraint formulation Mavrotas proposes in [1] to convert the inequality constraints to equality constraints by explicitly incorporating additional slack variables.

Lexicographic optimization for the payoff table

In order to properly apply the epsilon-constraint method we must have good estimate of the range of each objective function at least for the $p-1$ objective functions which will be used in the constraints. The calculation of the range is not trivial in terms of complexity and it may incur significant performance hit on the execution time of the algorithm using the epsilon constraint method. While the best value (the *ideal point* defined earlier) is easily attainable as optimal of the individual optimization, the worst value over the efficient set (the *nadir point* defined earlier) is not. The most common approach is to calculate these ranges from the *payoff table* (also defined earlier). Recall the payoff table is constructed by tabulating the results from the individual optimization of the p objective functions. The nadir value is usually approximated with the minimum of the corresponding column. However, even in this case we must be sure that the obtained solutions from the individual optimization of the objective functions are indeed Pareto optimal solutions. In the presence of alternative optimal the obtained solution is not guaranteed to be Pareto optimal using this method. In order to overcome this problem Mavrotas proposes (see [1]) the use of lexicographic optimization for every objective function in order to construct the payoff table with only Pareto optimal solutions. A simple remedy in order to bypass the difficulty of estimating the nadir values of the objective functions is to define reservation values for the objective functions. The reservation value acts like an upper bound in case of minimization of the objective functions. Values worse than the reservation values are not allowed.

In general the lexicographic optimization of series of objective functions is to optimize the first objective function and then among the possible alternative optima optimize for the second objective function and so on. The lexicographic optimization is performed as follows: we optimize the first objective function with highest priority) – in our case this is the *Delay* objective – for which we obtain $\max f_1 = z_1^*$. Then we optimize the second objective function by adding the constraint $f_1 = z_1^*$ in order to keep the optimal solution of the first optimization. Assume that we obtain $\max f_2 = z_2^*$. Subsequently we optimize the third objective adding the constraints $f_1 = z_1^*$ and $f_2 = z_2^*$ and so on.

After the calculation of the payoff table we divide the ranges of the objective functions into n equal intervals and we use $n + 1$ grid points as the values of $e_2, e_3, \dots, e_p$. In general this technique will result in a grid of size $\underbrace{n \times n \times \cdots \times n}_{p \text{ times}}$ and we iterate the values of e_i over the grid taking each one of the $n + 1$ values. Each grid segment can yield a possible non-dominated solution. The drawback of the grid-search method is that we will need to recompute the optimization problem n^p times. More on the grid based computation including an example for $p = 2$ can be found in Mavrotas' paper [1].

The augmented epsilon constraint methods of Mavrotas

The augmented version of the epsilon constraint formulation for multi objective optimization problems proposed by Mavrotas is (see [1]):

$$\min f_1(\mathbf{x}) + \varepsilon \times \left(\frac{s_2}{r_2} + \frac{s_3}{r_3} + \cdots + \frac{s_p}{r_p}\right) \qquad \textbf{(III)}$$

subject to:

$$f_2(\mathbf{x}) - s_2 = e_2$$
$$f_3(\mathbf{x}) - s_3 = e_3$$
$$\dots$$
$$f_p(\mathbf{x}) - s_p = e_p$$
$$\mathbf{x} \in \mathcal{X}, s_i \in \mathbb{R}_+, i = 1 \dots p, \left(f_1(\mathbf{x}), f_2(\mathbf{x}), f_3(\mathbf{x}), \dots, f_p(\mathbf{x})\right) \in \mathcal{Y} \equiv \mathbb{R}^p,\ \left(s_2, s_3, \dots, s_p\right) \in \mathbb{R}^{p-1}, \left(r_2, r_3, \dots, r_p\right) \in \mathbb{R}^{p-1}$$

Here $e_2, e_3, \dots, e_p$ are the parameters for the right hand side for the specific iteration drawn from the grid points of the objective functions $2,3, \dots, p$. The parameters $r_2, r_3, \dots, r_p$ are the ranges of the respective objective functions. $s_2, s_3, \dots, s_p$ are the additional slack variables corresponding to the respective constraints. The algorithm parameter ε should be sufficiently small; usually in the range $[10^{-6}, 10^{-3}]$.

Proposition: The formulation (III) of the epsilon-constraint method produces only efficient solutions (it avoids the generation of weakly-efficient solutions).

There is another augmented epsilon constraint method proposed by Mavrotas in [3] which makes enhancements in two aspects with respect to the epsilon constraint formulation (III):

a) computes the range of the objective functions over the efficient set by better calculations of the nadir values
b) modifies the extended objective function expression thereby enforcing the Delay to be the primary objective and lexicographic optimization for the rest of the objectives (shipping cost and CO2 emissions).

The augmented epsilon-constraint formulation defined in [3] will not be discussed here and we will limit our attention to the formulation (III) as well as the epsilon-constraint formulation discussed in Chapter 4 of [2]. The former will be denoted in our `FulOpt` code as the *Mavrotas' implementation* while the latter will be denoted as *Ehrgott's implementation*.

Formulation of the Epsilon-Constraint Algorithm for Fulfillment Optimization

In the developed multi-objective programming model, the goal is to provide a solution compromise among cost, CO2 emissions, and delay. The proposed model, in which cost, CO2 emission, and delay are minimized, can be summarized as follows:

$$\underset{s.t.\ \mathbf{x}\in S}{Min}\{TSC, TCO, DEL\}$$

Here $\mathbf{x}$ is the vector of decision variables, TSC, TCO and DEL are cost, CO2 emissions and delay objective functions, respectively, and S denotes the space of feasible solutions. In the ε constraint method, the original multi-objective problem is converted into a single objective programming model by retaining one of the objective functions as the primary objective function in the model. Accordingly, the remaining objective functions are expressed as the properly defined constraints with enforcing upper and/or lower bounds. Therefore, if delay objective function, i.e. DEL is chosen as the primary objective function, the following single objective programming model is attained where cost and CO2 emission objective functions are treated as model constraints:

$$\begin{aligned} Min \quad & DEL \\ s.t. \quad & TSC \le TSC_{min} + v\Delta\varepsilon_{TCS} \\ & TCO \le TCO_{min} + v\Delta\varepsilon_{TCO} \\ & \mathbf{x} \in S \end{aligned}$$

where $v = 0,1,\dots,\vartheta$, $\Delta\varepsilon_{TCS} = \frac{TSC^{max} - TSC^{min}}{\vartheta}$ and $\Delta\varepsilon_{TCO} = \frac{TCO^{max} - TCO^{min}}{\vartheta}$. In order to derive the maximum and minimum values of cost and CO2 emission objective functions, the following steps are taken:

- Obtain the global optimum solution of each objective function over S. Then, let $\mathbb{X} = \{X^*_{TSC}, X^*_{TCO}, X^*_{DEL}\}$ are the obtained global optimum solutions corresponding to TSC, TCO and DEL objective functions.
- Find the values of the objective functions TSC and TCO at each point of $\mathbb{X}$.
- Compute the minimum values of TSC and TCO objective functions as $TSC^{min} = min\{TSC(x), x \in \mathbb{X}\}$ and $TCO^{min} = min\{TCO(x), x \in \mathbb{X}\}$.
- Compute the maximum values of TSC and TCO objective functions as $TSC^{max} = max\{TSC(x), x \in \mathbb{X}\}$ and $TCO^{max} = max\{TCO(x), x \in \mathbb{X}\}$.

However, to guarantee an optimum solution of current single objective problem is a Pareto optimal solution of the original multi-objective problem, the modification proposed by (Mavrotas, 2009) is applied here, where the constraints associated with the added functions are transformed to equality by explicitly incorporating the appropriate slack variables and then penalizing the current new variables in the single objective function. Thus we have:

$$\begin{aligned} & Min\ DEL + \theta(S_1 + S_2) \\ & s.t.\ TSC + S_1 = TSC_{min} + v\Delta\varepsilon_{TSC} \\ & \qquad TCO + S_2 = TCO_{min} + v\Delta\varepsilon_{TCO} \end{aligned}$$

where θ is an adequately small value that does not affect the objective function.

Literature

[1] Effective implementation of the e-constraint method in Multi-Objective Mathematical Programming problems, G. Mavrotas, 2009
[2] Multi-criteria Optimization, Matthias Ehrgott, Chapter 4, 2nd edition, 2005
[3] AUGMECON2: A novel version of the ε-constraint method for finding the exact Pareto set in Multi-Objective Integer Programming problems, G. Mavrotas et al, 2013