

---

# Erratum to: A copula-based heuristic for scenario generation

Michal Kaut

SINTEF Technology and Society, Trondheim, Norway

in: Computational Management Science. See also `BIBTEX` entry below.

---

`BIBTEX`:

```
@article{Kaut11Err,  
  author = {Michal Kaut},  
  title = {Erratum to: A copula-based heuristic for scenario generation},  
  journal = {Computational Management Science},  
  volume = {12},  
  number = {2},  
  pages = {341--343},  
  year = {2015},  
  doi = {10.1007/s10287-014-0224-8}  
}
```

© Springer-Verlag 2015.

The original publication is available at [www.springerlink.com](http://www.springerlink.com).

# Errata to ‘A copula-based heuristic for scenario generation’

Michal Kaut\*

October 2014

In the published version of the paper, the symbol  $\text{dev}(i, j)$  changes its definition in Section 3.2, without an explanation in the text. Moreover, equations (11) and (12), while being correct, are slightly different from those implemented in the code used to generate the test results presented in the paper.

In this errata, we present a corrected version of Section 3.2.

## 3.2 Heuristic for the average-deviation problem

In this section, we present a heuristic for the bivariate case of the average-deviation problem. Just like the MIP formulation (9) from the previous section, the heuristic works on the ranks of the copula and uses the difference between the cdfs as a measure of distance between the sample and the target. It is based on the observation from Eq. (2) that the rank cdf  $C_r(i, j)$  depends only on points  $\{(i', j') : i' \leq i \ \& \ j' \leq j\}$  of the grid. This means that we can construct the sample ‘row-wise’: for each rank  $j$ , we compute the deviation caused by its pairing to rank  $i$ , as the sum of absolute values of deviations at grid points  $(\cdot, j)$ . We denote this deviation by  $\text{dev}(i, j)$ , which consequently *changes its meaning from now on*:

$$\text{dev}(i, j) = \sum_{l=1}^S |C_r(l, j) - C_r^*(l, j)|. \quad (11)$$

We evaluate  $\text{dev}(i, j)$  for all unused ranks  $i$  and then choose the one with the smallest deviation. The resulting greedy heuristic is presented in Fig. 3.

The most time-critical part of the heuristic is the calculation of  $\text{dev}(i, j)$  on line 4: if we simply use the definition from Eq. (11), the calculation would require  $\mathcal{O}(S)$  operations, making the whole heuristic  $\mathcal{O}(S^3)$ . Fortunately, this can be improved upon using the fact that, for  $i > 1$ , we have

$$C_r(i, j) = C_r(i, j - 1) + \begin{cases} \frac{1}{S} & \text{if we couple } j \text{ to some } i' \leq i \\ 0 & \text{otherwise,} \end{cases}$$

---

\*SINTEF Technology and Society, Trondheim, Norway; [michal.kaut@sintef.no](mailto:michal.kaut@sintef.no).

which, in combination with (11), gives

$$\begin{aligned} \text{dev}(i, j) &= \sum_{l=1}^S |C_r(l, j-1) + \text{if}(l \geq i, \frac{1}{S}, 0) - C_r^*(l, j)| \\ &= \sum_{l=1}^{i-1} |C_r(l, j-1) - C_r^*(l, j)| + \sum_{l=i}^S |C_r(l, j-1) + \frac{1}{S} - C_r^*(l, j)| \end{aligned}$$

and therefore

$$\text{dev}(i-1, j) = \sum_{l=1}^{i-2} |C_r(l, j-1) - C_r^*(l, j)| + \sum_{l=i-1}^S |C_r(l, j-1) + \frac{1}{S} - C_r^*(l, j)|.$$

Subtracting the two then gives

$$\begin{aligned} \text{dev}(i, j) &= \text{dev}(i-1, j) + |C_r(i-1, j-1) - C_r^*(i-1, j)| \\ &\quad - |C_r(i-1, j-1) + \frac{1}{S} - C_r^*(i-1, j)|, \end{aligned} \tag{12}$$

which is then initialized with

$$\text{dev}(0, j) = \sum_{l=1}^S |C_r(l, j-1) + \frac{1}{S} - C_r^*(l, j)|.$$

This recursive formula allows us to compute  $\text{dev}(i, j)$ , except  $\text{dev}(0, j)$ , in a constant number of operations, so the heuristic becomes  $\mathcal{O}(S^2)$ —a big difference to the exponentially growing solution time of the MIP formulations. In practice, this means that we can generate five thousand samples in less than a second on the same hardware we used for the MIP models. Note that the recursion implies that we have to evaluate  $\text{dev}(i, j)$  for all  $i \in \{1, \dots, S\}$ , instead of using only  $i \in \mathcal{I}$  as we do in the algorithm presented in Fig. 3.