A GPU-based Solution for Accelerating Spatially-Explicit Uncertainty- and Sensitivity Analysis in Multi-Criteria Decision Making

Christoph Erlacher[*1], Seda Şalap-Ayça [2], Piotr Jankowski[2], Karl-Heinrich Anders[1], Gernot Paulus[1]

[1] Carinthia University of Applied Sciences – Department of Geoinformation and Environmental Technologies, Austria
[2] San Diego State University – Department of Geography, United States of America

*Corresponding author: c.erlacher@cuas.at

Abstract
This study presents a GPU-based approach to accelerate the time consuming parts of the variance-based spatially-explicit Uncertainty and Sensitivity Analysis (U-SA). Performance comparisons were conducted in respect to a CPU-based NumPy and a GPU-based CUDA implementation. Preliminary results reported herein suggest that the proposed approach will provide a quantitative decision quality measure in complex and comprehensive spatial multi-criteria decision making processes and will allow reasonable computational times making spatially explicit variance-based SA applicable and attractive for large-size problems. Furthermore, it will be beneficial for different application domains like natural hazard risk assessment, landscape assessment, infrastructure planning, environmental impact assessment or identification of land use strategies for sustainable regional development.

Keywords
Spatial Decision Support, GPU, CUDA, Parallelization, Spatially-Explicit Uncertainty- and Sensitivity Analysis, Multi-Criteria Decision Making, MCDM.

1. Introduction

Uncertainty and Sensitivity Analysis (U-SA) is a critical step in multi-criteria decision making. The major goal is to verify the robustness and stability of an implemented model with respect to the existing uncertainties. Saltelli et al. (2008) define sensitivity analysis as "*the study of how uncertainty in the output of a model (numerical or otherwise) can be apportioned to different sources in the model input*". Uncertainties can refer to the criterion weights, the decision rules, the standardization process of the input criteria and the accuracy as well as the resolution of the input data (Hwang and Yoon, 1981; Ligmann-Zielinska and Jankowski, 2012; Ganji et al., 2016). Only a handful studies focused on spatially-explicit U-SA of model input factors (Ligmann-Zielinska and Jankowski, 2012; Ligmann-Zielinska and Jankowski, 2014; Şalap-Ayça and Jankowski 2016). As stated by Feizizadeh et al. (2014), Ligman-Zielinska and Jankowski (2014) and Şalap-Ayça and Jankowski (2016) spatially-explicit uncertainty and sensitivity analysis can be a time consuming process, which depends on the project area, the number of criteria and models runs associated with the sample size.

Therefore, this paper reports on the study of high-performance computing approach for spatially-explicit U-SA of the assessment of agricultural land units with a multiple-criteria decision making model. Two implementations, a CPU-based NumPy and a GPU-based CUDA (Compute Unified Device Architecture) solution, are compared regarding the computational time of performing the simulations. The Environmental Benefit Index (EBI) is used by the United States Department of Agriculture (USDA) to prioritize agricultural land units based on environmental benefits and select high-scoring units for USDA crop reserve program. This simple scoring model has been used by the Farm Service Agency (FSA) since 1990 to rank farmers' requests to enroll land into the Conservation Reserve Program during each general sign-up period (competitive bidding). The model includes five environmental factors at the top level ("Wildlife", "Water-Quality", "Soil-Erosion", "Enduring-Benefits" and "Air-Quality"). Detailed information concerning the EBI-Framework can be found in Şalap-Ayça and Jankowski (2016).

2. Theoretical Background

2.1. Multi-Criteria Decision Making

Multi-Criteria Decision Making (MCDM) techniques (Malczewski, 1999; Malczewski, 2006; Malczewski and Rinner, 2015) belong to the field of Spatial Decision Support Systems (SDSS) and consider a set of alternatives that are evaluated on the basis of conflicting criteria. MCDM problems include several objectives and each objective is expressed by a set of criteria where every criterion refers to a certain influence value that is indicated by the criterion weight. The combination of MCDM and Geographic Information Systems (GIS) facilitates the integration of the spatial aspects with MCDM. Geographical data are expressed as criterion maps, which can be differentiated into factor and constraint maps.

In short, the workflow of the decision making process includes the standardization of the criteria (transformation) regarding comparability and the application of decision rules like Weighted Linear Combination (WLC), Ordered Weighted Averaging (Yager, 1988; Malczewski, 2006) or Analytic Hierarchy Process (Saaty, 1980), which results in the monotonic rank-order of decision alternatives leading to final recommendation.

2.2. Spatially-Explicit Uncertainty and Sensitivity Analysis

As previously mentioned, sensitivity analysis is a critical part of MCDM in order to verify the robustness and stability of the implemented model. In spatial multi-criteria evaluation uncertainty can be potentially associated with the selection of decision criteria, criteria measurement (inaccuracy and errors) and expert's preferences that represent the criteria weights. Ligmann-Zielinska and Jankowski (2008) state, that a decision situation typically comprises aspatial as well as spatial aspects. However, the vast majority of reported sensitivity analysis studies and evaluation methods concern only the aspatial nature of decision situations while spatial characteristics are only included implicitly. Due to the fact that the spatial distribution of options and their criteria values may potentially influence the rank order of alternatives, the authors emphasize the consideration of spatial criteria (e.g. *proximity, compactness, contiguity, etc.*) and spatial weighting (varying the criterion importance over space and assigning different weights to different spatial units) in GIS-based multi criteria evaluation, which calls for new, spatially-explicit methods of U-SA. As described by Ligmann-Zielinska and Jankowski (2014) uncertainty analysis helps to quantify the variability of model outcomes, whereas sensitivity analysis focuses on the identification of decision criteria or criteria weights that are responsible for the variability. Ligmann-Zielinska and Jankowski (2014) examined the robustness of land suitability evaluation with help of Monte Carlo Simulation (MCS) and variance-based sensitivity analysis. Variance-based sensitivity analysis is often recommended because it represents a model independent procedure and is applicable for spatially-explicit data. Furthermore, in contrast to one at a time SA (Si, first order sensitivity indices) variance-based SA also incorporates the interaction of input factors (ST, total order sensitivity indices). *The (S,ST) pair offers a succinct yet comprehensive measure of input influence* (Ligmann-Zielinska and Jankowski, 2014).

2.3. GPU-based Parallelization

In the course of this research project, a GPU-based prototype was developed focusing on an acceleration strategy for performing the MCS. MCS represents the most time consuming component of the variance-based spatially-explicit uncertainty and sensitivity analysis method, which depends on the number of criteria and model runs associated with the sample size. Tang and Jia (2014) present an approach that focuses on parallel Graphics Processing Units (GPUs) in order to enable and accelerate the sensitivity analysis of large agent- based modelling of spatial opinion exchange. A solution for accelerating time-consuming calculations is domain-specific and depends on the implemented mathematical functions. General-Purpose Computing on Graphics Processing Units (GPGPUs) are used for computer graphics and entertainment, or GPU co-processors that are constructed to accelerate massively parallel floating-point operations (Krömer et al, 2014). The GPU accelerated computing prototype utilizes the CUDA architecture for NVIDIA GPUs. CPUs (Central Processing Unit) consist of few cores and are optimized for sequential serial processing. GPUs have a parallel architecture that incorporates thousands of smaller but more efficient cores. The calculations are organized into threads, blocks and grids, and are distributed with respect to the streaming multiprocessors and cores. Detailed explanations on GPU programming can be found in Wilt (2013).

3. Methodology

The original spatially-explicit Sensitivity and Uncertainty Framework (Ligmann-Zielinska and Jankowski, 2014) refers to a Python implementation that incorporates two dimensional NumPy arrays for the weight samples (N) and the criterion maps (k). The weight samples are generated using the quasi-random Sobol's experimental design (Software Simlab) and the Ideal Point (IP) decision rule is used to compute the suitability surfaces (see Figure 1). The total number of model runs (R) is represented by the following formula:

$$R = (k + 2) * N \tag{1}$$

In order to compare the performance of the original Framework with the CUDA-GPU version two-dimensional NumPy arrays representing evaluation criteria are flattened in order to calculate the suitability surfaces (see Figure 2). Consequently, the dimension of the final suitability surface is represented by the number of model runs (rows) and the number locations (raster cells). Each row of the final suitability surface represents one suitability map that is associated with the corresponding weight sample and the used decision rule (see Figure 3).

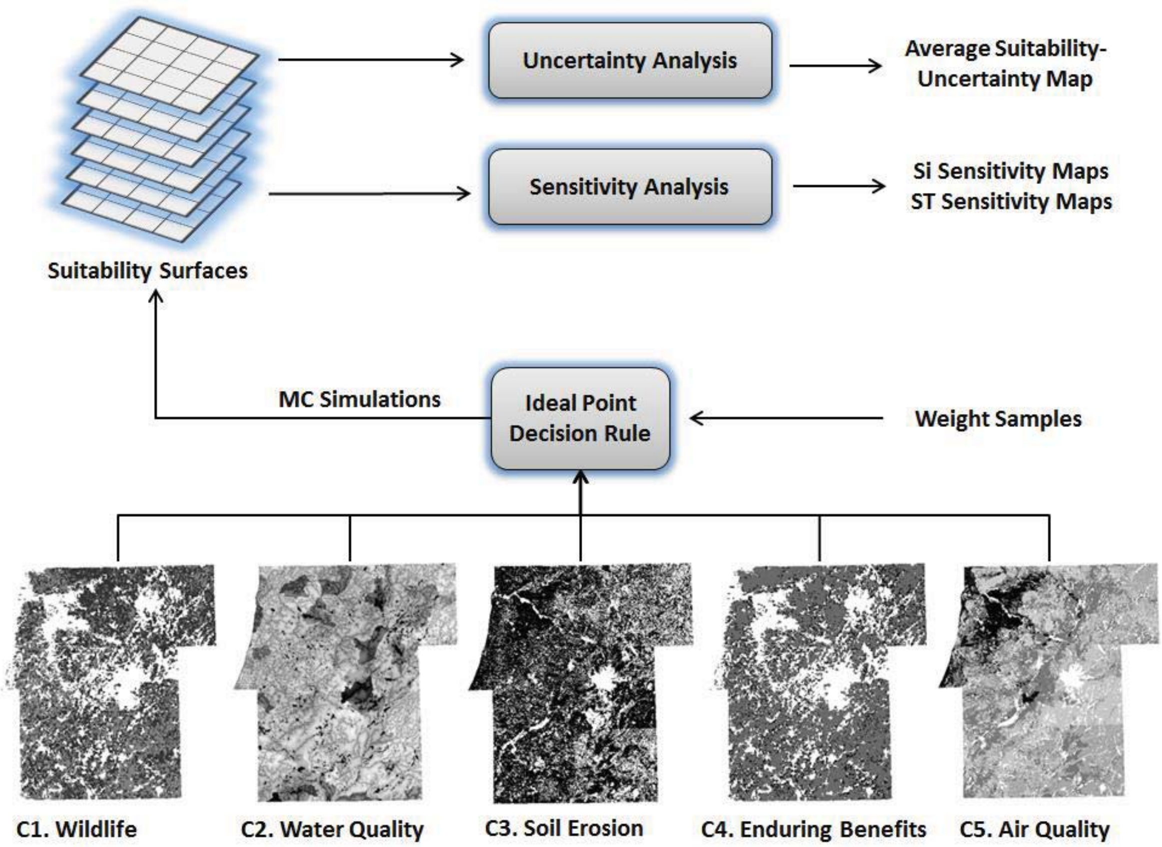

Figure 1: Displays the workflow to conduct a variance-based spatially-explicit U-SA.

Criterion 1

0.7	0.1	0.0
0.5	0.2	0.6
0.8	1.0	0.3

Criterion 2

0.2	0.1	0.0
0.4	0.7	0.3
0.6	1.0	0.4

Criterion 3

0.5	0.2	0.0
0.4	0.7	0.3
0.6	1.0	0.4

Criterion 4

0.2	0.1	0.0
0.4	0.7	0.3
0.6	1.0	0.4

Criterion 5

0.2	0.1	0.0
0.4	0.7	0.3
0.6	1.0	0.4

Weights

Criterion 1	0.2
Criterion 2	0.15
Criterion 3	0.1
Criterion 4	0.3
Criterion 5	0.25

Flatten Criterion Matrix

0.7	0.1	0.0	0.5	0.2	0.6	0.8	1.0	0.3
0.2	0.1	0.0	0.4	0.7	0.3	0.6	1.0	0.4
0.5	0.1	0.0	1.0	0.2	0.8	0.9	0.1	0.4
0.2	0.3	0.0	0.6	0.2	0.9	0.3	0.5	0.1
0.1	0.4	0.0	0.2	0.6	0.5	0.8	0.2	0.6

Simulation Output
Based on Decision Rule SAW

0.305	0.245	0	0.49	0.375	0.64	0.75	0.56	0.34

Figure 2: Represents the data structure to perform the GPU-based MCS.

Weight Sample

#1	0.2	0.15	0.1	0.3	0.25
#2	0.25	0.1	0.15	0.2	0.3
#3	0.15	0.2	0.2	0.25	0.2
#4	0.3	0.15	0.15	0.2	0.2
#...	...	...	....		

Output of Simulations

#1	0.45	0.1	0	0.72	0.35	0.61	0.79	0.55	0.38
#2	0.35	0.245	0	0.495	0.37	0.63	0.775	0.525	0.375
#3	0.315	0.23	0	0.545	0.38	0.635	0.775	0.535	0.35
#4	0.375	0.215	0	0.52	0.355	0.625	0.765	0.605	0.35
#..	...	...	...	...	...	...	...	...	...

Figure 3: Illustrates the structure of the suitability surfaces, where each row represents one suitability raster map based on the decision rule and the criterion weights.

The algorithm to perform the MCS for the CUDA-GPU implementation is divided into the host part (CPU) and the device part (GPU). Parallel portions of the application are executed on the device as kernels, whereas the host represents the program logic for memory allocation, data partitioning and recombination (see Figure 4). The data partitioning depends on the computing and memory capabilities of the device in order to avoid memory overflow. Therefore, the first task of the controller is to identify the available computing capacity of the device. For example, GPU memory could be allocated by another software or task. According to the determined GPU resources the number of partitions is calculated. For each partition the device memory for a subset of the suitability surfaces has to be allocated, which is necessary for calling kernels. A kernel is launched as a grid of thread blocks and each thread block consists of a specified number of threads (e.g. warp size). The number of blocks per grid and the number of threads per block depend on the number of rows and columns of the suitability surface.

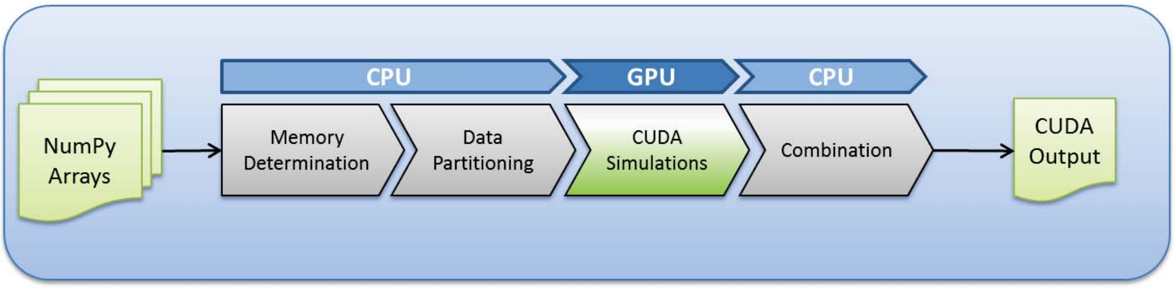

Figure 4: Overview about the CUDA-GPU based MSC computation.

Figure 5 illustrates a more detailed representation of the CUDA-GPU based Monte Carlo Simulations to generate the suitability surfaces.

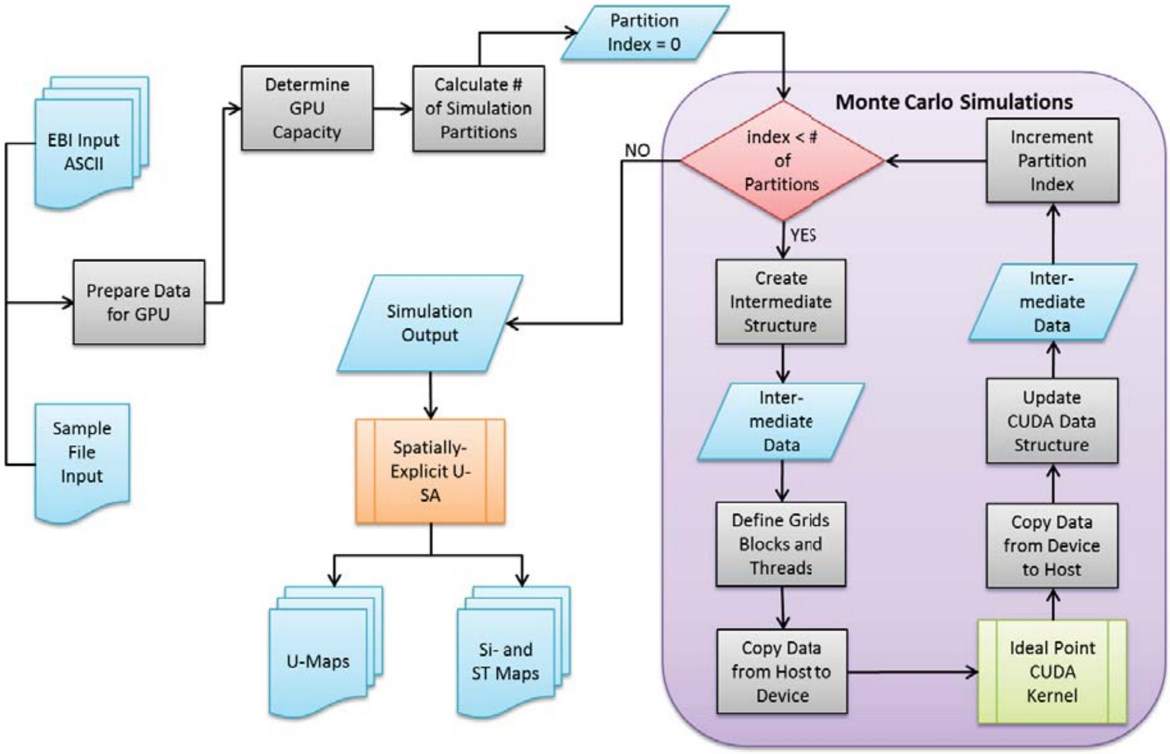

Figure 5: Detailed workflow of the CUDA-GPU based MSC computation.

4. First Speed Up Results

In order to facilitate the performance tests, different NVIDIA GPUs were investigated. The results confirm the efficacy of employing GPU-based solutions for computationally demanding, spatially-explicit U-SA. For example, for the Tesla k40c device, a computational acceleration up to a factor of 150 was achieved. The case study tested model inputs with 5 input factors, 1,475,464 locations, and 2,664 simulation runs.

Tables 1-3 illustrate CPU- and GPU specification overview and the performance comparison of different GPUs as well as a performance comparison regarding the CPU (NumPy – a package for scientific computing with Python) and CUDA (Anaconda NumbaPro) implementation.

Tesla – GPU Specifications	Tesla K40 Workstation	Tesla K20m Server
Peak double-precision floating point performance (Tflops)	1.43	1.17
Peak single-precision floating point performance (Tflops)	4.29	3.52
# of CUDA cores	2880	2496
Total Global Memory – GDDR5 (GB)	12	5
Memory bandwidth (GB/sec)	288	208

Table 1: Tesla GPU Specification Overview.

CPU Specifications	Intel Xeon E5-1620 v3 Workstation	Intel Xeon E5-2697 v2 Server
Intel Smart Cache (MB)	10	30
# of Cores	4	12
# of Threads	8	24
Processor Base Frequency (GHz)	3.5	2.7
Max memory bandwidth (GB/sec)	68	59.7

Table 2: Intel Xeon CPU Specification Overview.

Table 3 reports the relative speed ups for both Tesla GPU cards with respect to the computational demand of the CPUs. The Tesla K40c device (Workstation) is 1.809 times faster than the Tesla K20m device (Server).

Performance Comparison	Workstation	Server
Avg. Elapsed Time for CUDA (min)	0.1215	0.2198
Avg. Elapsed Time for NumPy (min)	18.6712	25.7937
CUDA Speed Up	153.6724	117.3508

Table 3: CUDA-NumPy Performance Comparison.

5. Discussion and Future Prospects

The implemented GPU-based prototype offers reasonable computation times to perform spatially-explicit U-SA, which makes this type of sensitivity analysis applicable and potentially attractive for distributed-output (spatially-explicit) models without using expensive servers or super computers. Furthermore, this solution allows the integration of further decision rules (WLC, OWA etc.) by changing the kernel function and can be applied for different application areas.

The proposed workflow incorporating the data preparation regarding the supported interface and the partitioning and recombination of the GPU-based computation is limited to local operations. For supporting the calculation of local variability (focal- and zonal operations) in respect to criteria values and weights, a conceptual definition for the data structure as well as partitioning- and recombination approach have to be adapted (Malczewski, 2011; Şalap-Ayça and Jankowski; 2016).

Further accelerations can be achieved by implementing additional CUDA kernels for the uncertainty and sensitivity computations. Additionally, the distribution of the CUDA calculations among many GPUs will be investigated too as the next step of reported here research.

6. Acknowledgement

Financial support for this work was provided by the National Science Foundation Geography and Spatial Sciences Program Grant No. BCS-1263071. Any opinion, findings, conclusions, and recommendations expressed in this material are those of the authors(s) and do not necessarily reflect the views of the National Science Foundation.

References

Feizizadeh, B., Jankowski, P., Blaschke, T. (2014). A GIS based Spatially-explicit Sensitivity and Uncertainty Analysis Approach for Multi-Criteria Decision Analysis. *Computers and Geosciences*, Volume 64, pp. 81-95.

Ganji, A., Maier, H., Dandy, G. (2016). A modified Sobol' sensitivity analysis method for decision-making in environmental problems. *Environmental Modelling & Software*, Volume 75, pp. 15-27.

Hwang, C.L., Yoon, K., 1981. Multiple Attribute Decision Making Methods and Applications: A State of the Art Survey. Springer-Verlag, Berlin.

Krömer, P., Platoš, J., Snášel, V. (2014). Nature-Inspired Meta-Heuristics on Modern GPUs: State of the Art and Brief Survey of Selected Algorithms. *International Journal of Parallel Programming,* Volume 42, Issue 5, pp. 681–709.

Ligmann-Zielinska, A., Jankowski, P. (2014). Spatially-Explicit Integrated Uncertainty and Sensitivity Analysis of Criteria Weights in Multicriteria Land Suitability Evaluation. *Environmental Modelling & Software*, Volume 57, pp. 235-247.

Ligmann-Zielinska, A., Jankowski, P. (2012). Impact of proximity-adjusted preferences on rank-order stability in geographical multicriteria decision analysis. *J. Geogr. Syst*. Volume 14, Issue 2, pp. 167-187.

Ligmann-Zielinska, A., Jankowski, P. (2008). *A Framework for Sensitivity Analysis in Spatial Multiple Criteria Evaluation*. In: T.J. Civa et al., eds. GISience 2008, LNCS 5266, Springer-Verlag Berlin, pp. 217-333.

Malczewski, J. (1999). GIS and Multicriteria Decision Analysis. New York, Wiley.

Malczewski, J. (2006). Ordered weighted averaging with fuzzy quantifiers: GIS-based multicriteria evaluation for land-use suitability analysis. *International Journal of Applied Earth Observation and Geoinformation*. Volume 8, pp. 270–277.

Malczewski, J. (2011). Local Weighted Linear Combination. *Transactions in GIS*, Volume 15, Issue 4, pp. 439-455.

Malczewski, J., Rinner, C. (2015). Multicriteria Decision Analysis in Geographic Information Science, Advances in Geographic Information Science. New York, Springer.

Saaty, T.-L., 1980. The analytic hierarchy process. New York, McGraw-Hill.

Şalap-Ayça, S., Jankowski, P. (2016). Integrating Local Multi-Criteria Evaluation with Spatially Explicit Uncertainty-Sensitivity Analysis, Spatial Cognition & Computation, DOI: 10.1080/13875868.2015.1137578

Saltelli, A., Ratto, M., Andres, T., Campolongo, F., Cariboni, J., Gatelli, D., Saisana, M., Tarantola, S. (2008). *Global Sensitivity Analysis*: The Primer. John While & Sons, Ltd.

Tang, W., Jia, M. (2014). Global Sensitivity Analysis of a Large Agent-Based Model of Spatial Opinion Exchange: A Heterogeneous Multi-GPU Acceleration Approach. *Annals of the Association of American Geographers*, Volume 104, Issue 3, pp. 485-509.

Wilt, N. (2013). The CUDA Handbook: A Comprehensive Guide to GPU programming. New Your, Addison-Wesley.

Yager, R.-R. (1988). On ordered weighted averaging aggregation operators in multi-criteria decision making. *IEEE Transactions on Systems, Man and Cybernetics, Volume* 18, Issue 1, pp. 183-190, IEEE Press Piscataway, NJ, USA.