

Data Mining Approaches for Life Cycle Assessment

Naren Sundaravaradan, Manish Marwah, Amip Shah, Naren Ramakrishnan

Abstract—To reduce the cost associated with screening lifecycle assessments (LCAs), we propose treating LCA as a data mining problem and automating the process of assigning impact factors to inventory components. We describe three sub-problems involved in automating LCA and provide illustrative case studies for each. The results from an automated data mining approach are comparable to those obtained from the more laborious manual LCA process.

Index Terms—Data mining, environmental management, green design, environmental factors

I. PROBLEM ADDRESSED

Screening life-cycle assessments (LCAs) [1-2] are of interest in identifying the environmental “hotspots” which need to be mitigated for different products [3-4]. Often, however, especially for new products which are yet to be fabricated, LCA practitioners have no information about the product inventory other than a bill-of-materials (BOM) which lists the components that go into creating the system. Current practice is to try and manually map each component of a given BOM to a functional unit within a published environmental database [10-15], but the cost entailed in such manual mapping is often prohibitive [5-9].

We propose treating screening LCA as a data mining problem, with the lifecycle inventory being set up as a product tree; an environmental database being set up as a matrix; and the impact assessment being set up as a similarity problem. Using such a formulation, we show how analytical techniques can be leveraged to address a variety of problems related to screening LCAs.

II. KNOWLEDGE OF PRIOR WORK

Most prior work related to data analytics in LCA has occurred in the context of uncertainty analysis, sensitivity studies, and data quality assessments. These include statistical methods to understand impact variation [16-19], simulations (such as Monte Carlo) to test robustness [20-26], and empirical methods to evaluate the quality of data used within

the LCA [27-32]. There is also prior work related to the application of weighting schemes to determine the appropriate normalization and characterization of impact categories [7, 16, 32]. However,, we are not aware of any prior work related to the application of quantitative data mining techniques for simplifying inventory management or impact assessment within an LCA.

III. PROJECT UNDERTAKEN

This paper describes an investigation into opportunities for leveraging the growing field of ‘data mining’. Specifically, we illustrate the applicability of data mining in three areas:

(i) filling in missing impact data or validating existing data within published environmental databases, so that the integrity of impact assessment may be improved and the cost of creating such databases may be reduced (“matrix completion”);

(ii) reconstructing system boundaries and lifecycle inventory (LCI) for published LCA studies where only endpoint values are reported but no inventory data are provided (“tree discovery”); and,

(iii) expeditious assessment of environmental footprint by matching a given lifecycle inventory component to similar environmental database nodes (“node similarity”).

When combined, the above three methods provide a means to simplify and automate the process of performing screening LCA studies, thus greatly reducing the cost of performing hotspot analysis across a large range of products.

IV. RESEARCH METHODS

Data mining [33] denotes a broad range of statistical and artificial intelligence techniques applied to large scale data with the goal of automatically extracting knowledge. In addition to commonly used techniques such as regression, data mining includes techniques and algorithms for performing tasks such as classification, clustering and association rule mining [34]. Specifically in this paper, data mining techniques used include clustering¹, k -nearest neighbors classification², and non-negative least squares regression³.

¹ Clustering involves assigning data points to groups called clusters based on a distance metric (e.g. Euclidean distance) between the data points.

² k -nearest neighbors classification assigns data points to classes or groups based on k other data points that are most similar to it (based on a distance metric).

³ Non-negative least squares regression is just like ordinary least squares regression with the added constraint that all coefficients to be determined must be non-negative.

This work was partly supported by a grant from the Hewlett Packard Innovation Research Program.

M. Marwah and A. Shah are with the Sustainable Ecosystems Research Group at Hewlett Packard Laboratories, Palo Alto, CA 94304 (e-mail: manish.marwah@hp.com; amip.shah@hp.com).

N. Sundaravaradan and N. Ramakrishnan are with the Department of Computer Science, Virginia Polytechnic Institute and State University (Virginia Tech), Blacksburg, VA 24061 (e-mail: narens@vt.edu; naren@vt.edu).

In order to make environmental data more amenable to data mining techniques, we structure a given environmental impacts dataset as a matrix as shown in Fig. 1(a). The rows of the matrix correspond to functional units in the environmental database while the columns correspond to impact factors. Next, the LCI is rearranged as a ‘tree’ (Fig. 1(b)) where the topmost (root or parent) node is the system to be analyzed, and subsequent nodes (children) are descriptions of how the different parts within the system are related.

With the above structure, it becomes possible to borrow numerous algorithmic approaches from the world of data mining.

	Impact Factor I_1	Impact Factor I_2	Impact Factor I_3	Impact Factor I_4	...
Functional Unit 1	$I_{1,1}$	$I_{1,2}$	$I_{1,3}$	$I_{1,4}$...
Functional Unit 2	$I_{2,1}$	$I_{2,2}$	$I_{2,3}$	$I_{2,4}$...
Functional Unit 3	$I_{3,1}$	$I_{3,2}$	$I_{3,3}$	$I_{3,4}$...
...

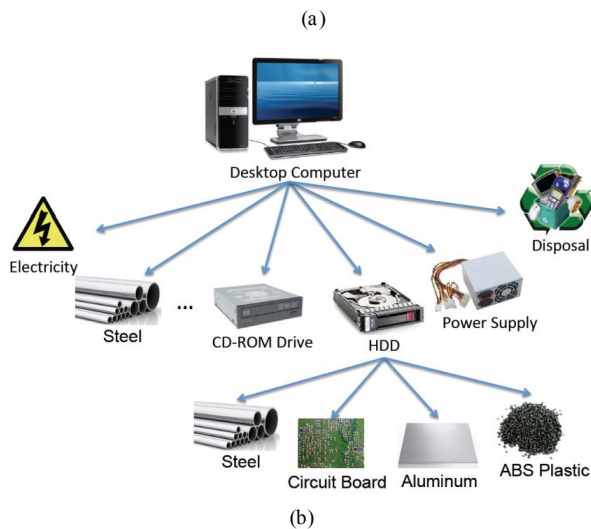


Figure 1. (a) Illustration of environmental impact database, set up as a matrix of impact factors. (b) Tree diagram for a desktop computer.

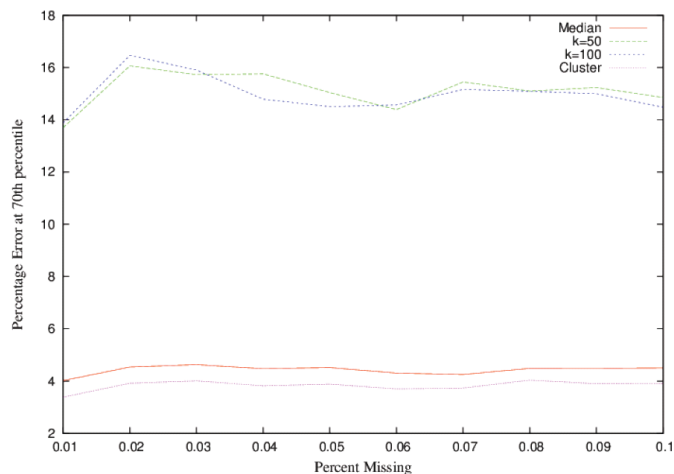


Figure 2. Reconstruction percentage error (70th percentile) is shown as a function of increasing percentage missing data (from 1% to 10%). The cluster method performs the best (3.5 ~ 4% error) across the percent missing range.

A. Matrix Completion

A large scale environmental impacts dataset, such as ecoinvent 2.0, arranged as a matrix as shown in Fig 1 (a), may contain missing, invalid, or unavailable entries. We refer to estimating these unknown values as the *matrix completion problem*. Missing entries could arise due to high cost or difficulty associated with estimating a specific impact factor for a particular node. In addition, when new nodes are to be added to an existing environmental database, it is usually expensive to evaluate all the environmental impact factors related to that particular node. For example, commercial off-the-shelf databases often have upwards of 200 impact factors. For each of these impact factors, a practitioner must manually evaluate the impact of the system in order to fill out the database. We present methods and techniques by which a practitioner could omit select impact factors during data input, and then automatically estimate these missing impact values at a later time. Such an approach could reduce the cost of creating large-scale environmental databases.

To illustrate this approach, we apply it to the ecoinvent 2.0 database. In order that ground truth be available to estimate the accuracy of our methods, we randomly remove up to 10% of the known impact factors, and then estimate these values using multiple matrix completion algorithms.

We apply three different methods for reconstructing the missing values. The first method fills in the missing values with the median of the corresponding impact factor derived from values that are present. The second method first calculates the k -nearest neighbors of each node and then replaces the missing values in a node by the median derived from its k -nearest neighbors. The third method uses an iterative method to estimate by first randomly initializing the missing values. Each subsequent iteration consists of clustering the nodes using k -means algorithm into a given number of clusters and then replacing missing values in nodes by the medoids of the cluster it belongs to. This process is continued until convergence, or until number of iterations has reached a threshold.

To measure how closely the reconstructed values match the original ones we compute the percentage error between the correct values and the reconstructed ones. We then applied the different methods on the data with various percentages of missing values ranging from 1 percent to 10 percent. The data consists of 3828 nodes and 198 impact factors, so 10 percent amounts to about 75000 missing values. Figure 2 tracks the error at the 70th percentile over the various runs. From the 50th to 60th percentile the methods perform comparably, but at this point the methods involving clustering and medians perform the best. While the error using these methods stays under 5% for 90% of the missing values (about 67500), for the remaining (about 7500) values the errors may be quite large (up to 70%); this is likely because the remaining missing values are quite dissimilar from those of any of the other nodes with known values, so no reasonable basis exists upon which the missing values can be predicted for these ~10% nodes. Thus, we find that this approach is most suitable in cases where a new node being added to the database bears some resemblance to the properties of existing nodes; in situations where a completely new system is being added, estimating missing values through matrix completion may not be an appropriate approach.

B. Tree Discovery

Often, LCA studies report the system studied, the database used, and the resulting midpoints and endpoints; but fail to report the actual inventory breakdown. For situations where all the impact factors are drawn from the same database, it becomes possible to ‘discover’ the tree (inventory) underlying the system. Specifically, the total impact of the entire system (\mathbf{I} , which has been reported) must equal the sum of impact factors for n functional units (child nodes) weighted by the appropriate quantities:

$$\mathbf{I} = w_1 \mathbf{I}_1 + w_2 \mathbf{I}_2 + \dots + w_n \mathbf{I}_n \quad (1)$$

Note that \mathbf{I} and \mathbf{I}_i are vectors and each consists of m impact factors. The challenge is to search for the n correct nodes (note that n must also be determined since it is not known) from the database corresponding to the parent and then determine $\{w_1, w_2, \dots, w_n\}$ which will result in the closest match of the known total impact \mathbf{I} . Successfully solving this would allow both the inventory and the coefficients to be identified, even without knowledge of the inventory components.

The algorithm, since the goal is to find a linear fit, at its core makes use of the non-negative least squares algorithm (NNLS) [36] to test how well certain subsets of nodes fit. Since a brute-force exploration of the entire space of subsets (in other words, examining all possible combinations of nodes in a database to find the child nodes) is intractable, the algorithm selectively samples what we refer to as “generators”. A generator is defined to be a subset of nodes such that it contains a minimal number of nodes that satisfies a minimum threshold error criterion when supplied to NNLS, so if any node is removed from this subset it will no longer satisfy the threshold error criterion. The first step of the discovery algorithm fixes a particular node and then samples a fixed number of generators that contain it. The process is

repeated for all nodes. These generators will give us an idea of whether or not a particular node occurs concurrently with other nodes. If we refer to these generator samples for each node as “profiles”, then the next stage of the algorithm clusters similar profiles together. For instance, if node A occurs frequently with nodes B,C and D, and if node B occurs frequently with nodes A, C and D, then the profiles of A and B will belong in the same cluster because are similar. In essence, nodes within a cluster can be considered to provide similar information. The final stage in the algorithm tries to prune nodes from clusters to greedily increase the quality of the fit of the remaining subset of nodes after which we are left with the final tree [37].

Figure 3 illustrates the results of this approach for a power supply unit from the ecoinvent 2.0 database: the only input provided was the total impact of the power supply unit (as published); the algorithm described above then explored and evaluated potential inventory components from the rest of the database, determined the weighting coefficients for those inventory components which allowed for a reasonable match, and rejected those components which could not be fit. While the results from such an approach are approximate and may not work in all scenarios, this provides a low-cost method to perform hotspot analysis for screening-type studies.

Another example is provided in Figure 4, for a LiC6 Electrode. It also shows an analysis of the errors of the discovered tree as compared to the reference (or actual) tree. It plots the difference between the percentage contribution of an impact factor (median and max) in the discovered node minus the percentage contribution in the reference node. For example, the difference in the median and maximum contribution in node 1056 is very low indicating that the weight assigned to this node by the algorithm matches very closely with the actual weight of the node. In almost all cases the error in the median contribution is low, while the error in the maximum contribution varies. This is, in general, attributable to the diversity of the impact factor values within a node. A negative difference shows that the algorithm assigned a very low weight compared to the actual node (including 0, a false negative); a positive difference shows that the algorithm assigned too much weight to a node, including the cases where a node that does not exist in the actual tree is reported (i.e. a false positive). The number of nodes correctly reported in this case is 5 out of 6 (although an additional node discovered was close enough to the missed node to be considered substitutable by a domain expert).

The low error in median contribution implies that for the majority of impact factors, the predicted weight of the node discovered is similar to that reported in the published database. A high error in the maximum contribution suggests the possibility that for a small number of select impact factors – such as trace elements – the tree discovery algorithm is not sufficiently detecting these issues. It is worth noting however that this is also a limitation of screening LCAs in the state-of-the-art as well; if an effective means of detecting such trace impacts in screening LCAs were to be made available, these could potentially be corresponding encoded into the discovery

algorithm to minimize the maximum error. Figure 5 shows the percentage errors for a CDROM. Again the median errors are very low.

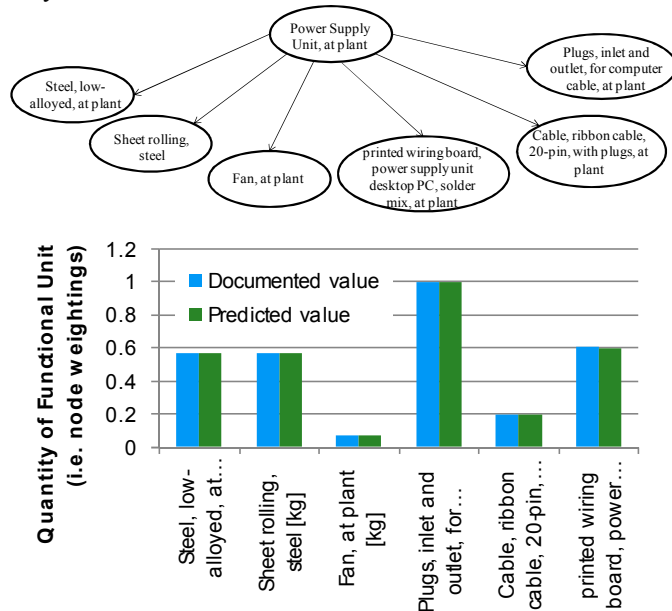


Figure 3. Comparison between predicted (from tree reconstruction) and documented (from published database) values of functional unit quantity for different constituent nodes of a power supply unit.

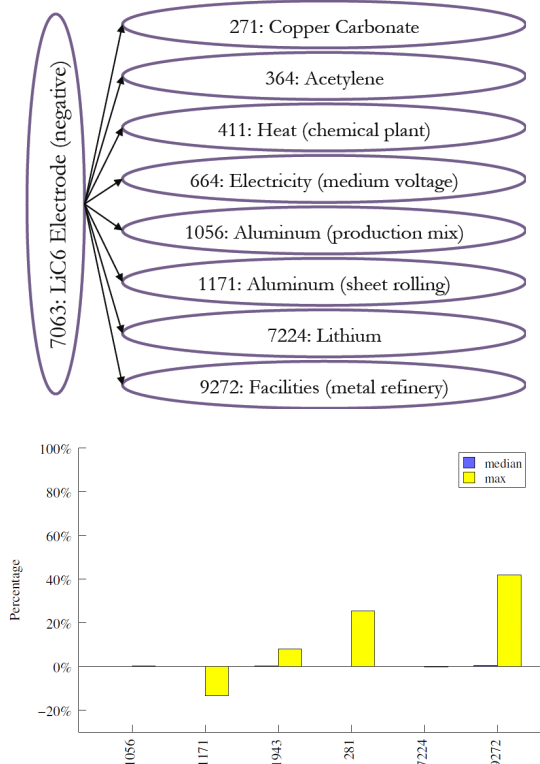


Figure 4. The top figure show the tree for LiC6 Electrode, while the bottom shows the median and maximum percentage error (across all the impact factors) in the discovered tree.

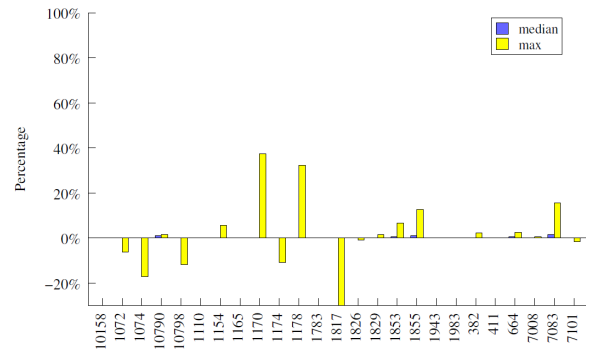


Figure 5. The median and maximum percentage errors of the discovered tree for a CDROM.

C. Node Similarity

Estimating product footprint typically involves manual disassembly of existing products, individually weighing each component, identifying the material content of each component, and then assigning environmental impact factors based on the material content plus mass. However, the large number of inputs required to accurately perform a life-cycle assessment limits the scalability of such an approach. For example, an IT system may have hundreds of material and energy flows across the supply chain. Manually tracking each of these flows is cumbersome and expensive. The ability to automate the process of environmental foot printing could be beneficial, both to reduce internal costs of foot printing as well as a services opportunity. First, we use *matrix completion*, as described earlier, to fill in any missing information in the environmental database used, followed by the following three steps.

Clustering. We perform node clustering for the hundreds of items listed in a particular BOM. This is done because many of the BOM components may be quite similar from the standpoint of environmental impact but very different in terms of how they are identified on the system tree (e.g., two identical stainless steel screws which reside in different parts of the system may have completely different part numbers). We propose automating this process through a ‘node similarity’ algorithm, where a quantitative similarity (or distance) metric is computed by comparing node specific information (such as, part name, description, etc.). For example, to compare text attributes of nodes, approximate string matching techniques such as longest common subsequence (LCS), longest common prefix (LCP), Levenshtein distance, or a combination thereof can be used [35]. Once a distance metric is obtained, clustering algorithms such as partitioning around medoids (PAM) can be used to group similar BOM nodes together. The resulting clusters reduce the number of parts to be evaluated from several thousand to a smaller and more manageable number (typically, at least an order of magnitude reduction).

Translation. Each of these clusters is then assigned a representative node that is similar from the environmental database. In other words, we ‘translate’ the different BOMs associated with a product (which may come from different suppliers and thus have different naming schemes) into a standard terminology, derived from the environmental

database, that allows us to gain insight into the environmental impact related to each cluster. Ideally, such translation could also be automated based on the above similarity techniques (e.g., where the BOM nodes are automatically assigned to environmental nodes with similar identifiers), but initially, we perform this translation manually. (It is worth noting that the clustering performed above is what allows such translation to be done manually, since clustering enables a reduction in the number of translations by an order of magnitude.)

Tree Reconstruction. One pending challenge with the translation is that the units specified on the BOM and environmental database may differ. For example, most product BOMs specify the number of repeating instances for a particular part number, while the environmental nodes may be specified per mass (kg) or such. To rectify this, we recognize that discrete environmental nodes within a tree comprise a set of simultaneous linear equations, because of the requirement that impacts of all the child nodes must approximately equal the impact of the root (parent). We propose a tree reconstruction algorithm that takes the disparate nodes (clusters) previously identified and performs a non-negative least squares (NNLS) fit to identify the appropriate coefficients (weights) for each node. With a single baseline allocation of the units (e.g. the total mass of a single instance of the root node from the BOM), it becomes possible to solve the above NNLS and reconstruct the environmental tree.

With the above building blocks in place, it becomes possible to obtain an approximate environmental footprint for any arbitrary tree or product BOM. We illustrate the above approach by analyzing a real printed circuit board (PCB) for an existing computer system. This PCB BOM contains about 560 components, including a mix of resistors, capacitors, ASIC and logic devices, etc. We begin by using the node similarity approach and clustering the BOM nodes using PAM with their text descriptions as attributes. We identify 22 unique clusters. Because a large number of components on the PCB are similar, it became relatively easy to translate these clusters into a list of nodes from the environmental database. For the resulting environmental tree, we were able to utilize the impact factors available from the environmental database and successfully solve for the coefficient of each node.

As proof-of-concept, we compare the total environmental footprint calculated using our analytics-based approach to one computed manually by an LCA practitioner. The results are quite encouraging. For the 3rd quartile (i.e. for nearly 75% of the impact factors, or about 180 different environmental impacts), the predictions agree to within 25%; with a median prediction difference of about 12.8% between the auto-predicted and manually calculated environmental footprint. Lastly, and perhaps most importantly, after having obtained the environmental footprint for each of the clusters, the results can be used to perform an environmental “hot spot” analysis, where essentially a Pareto list of the biggest environmental contributors to the overall PCB footprint is generated so that a designer or LCA practitioner can zoom in on where further efforts should be focused. The results from the auto-LCA Pareto match those obtained from a manual LCA: for

example, in terms of global warming potential (GWP), the chips are the highest footprint owing to their upstream manufacturing; followed by the copper in the connectors and the capacitors. Thus, a designer wanting to further reduce the environmental footprint of the PCB knows where efforts should be focused – and then the designer can use such a tool to reassess a new design as well. This is important because it becomes possible for someone who has no LCA or environmental background to automatically get feedback regarding the sustainability of their design.

V. CONCLUSIONS

Consumers and enterprises are seeking increasing amounts of environmental information about the products they purchase and produce. By applying data mining approaches to address three LCA problems – matrix completion, tree discovery, and node similarity – it becomes possible to significantly automate a screening LCA, while retaining reasonable levels of accuracy within fairly broad boundary constraints, especially for cases where the inventory data is of low quality. In the future, we plan to further evaluate the scalability of these algorithms and test the approaches on a broader variety of systems.

VI. REFERENCES

- [1] Baumann, H., Tillman, A.-M., 2004, *The Hitch Hiker’s Guide to LCA*, Studentlitteratur AB, Sweden.
- [2] Graedel, T. E., 1998, *Streamlined Life-Cycle Assessment*, Prentice Hall: New Jersey.
- [3] Glasson, J., Therivel, R., Chadwick, A., 2005, *Introduction to Environmental Impact Assessment: Principles and Procedures, Process, Practice and Prospects (3rd Edition)*, Taylor & Francis: New York.
- [4] Udo de Haes, H. A., Jolliet, O., Finnveden, G., Hauschild, M., Krewitt, W., Müller-Wenk, R., 1999, “Best Available Practice Regarding Impact Categories and Category Indicators in Life Cycle Impact Assessment,” *The International Journal of Life Cycle Assessment*, Vol. 4, No. 2, pp. 66-74.
- [5] Reap, J., Roman, F., Duncan, S., Bras, B., 2008, “A Survey of Unresolved Problems in Life-Cycle Assessment – Part 1: Goal and Scope and Inventory Analysis,” *The International Journal of Life-Cycle Assessment*, Vol. 13, No. 4, pp. 290-300.
- [6] Reap, J., Roman, F., Duncan, S., Bras, B., 2008, “A Survey of Unresolved Problems in Life-Cycle Assessment – Part 2: Impact Assessment and Interpretation,” *The International Journal of Life-Cycle Assessment*, Vol. 13, No. 5, pp. 374-388.
- [7] Pennington, D. W., Potting, J., Finnveden, G., Lindeijer, E., Jolliet, O., Rydberg, T., Rebitzer, G., 2004, “Life Cycle Assessment – Part 2: Current Impact Assessment Practice,” *Environment International*, Vol. 30, No. 5, pp. 721-739.
- [8] Finnveden, G., 2000, “On the Limitations of Life Cycle Assessment and Environmental Systems Analysis Tools in General,” *The International Journal of Life Cycle Assessment*, Vol. 5, No. 4, pp. 229-238.

- [9] Schaltegger, S., 1997, "Economics of Life Cycle Assessment: Inefficiency of the Present Approach," *Business Strategy and the Environment*, Vol. 6, No. 1, pp. 1-8.
- [10] Vigon, B. W., 1994, *Life-Cycle Assessment: Inventory Guidelines and Principles*, CRC Press: New York.
- [11] Jiménez-González, C., Kim, S., Overcash, M. R., 2000, "Methodology for Developing Gate-to-Gate Life Cycle Inventory Information," *The International Journal of Life Cycle Assessment*, Vol. 5, No. 3, pp. 153-159.
- [12] Owens, J. W., 1997, "Life-Cycle Assessment: Constraints on Moving from Inventory to Impact Assessment," *Journal of Industrial Ecology*, Vol. 1, No. 1, pp. 37-49.
- [13] Suh, S., Huppel, G., 2005, "Methods for Life Cycle Inventory of a Product," *Journal of Cleaner Production*, Vol. 13, No. 7, pp. 687-697.
- [14] Rebitzer, G., et al., 2004, "Life Cycle Assessment – Part 1: Framework, Goal and Scope Definition, Inventory Analysis, and Applications," *Environment International*, Vol. 30, No. 5, pp. 701-720.
- [15] Ekvall, T., Weidema, B. P., 2004, "System Boundaries and Input Data in Consequential Life-Cycle Inventory Analysis," *The International Journal of Life Cycle Assessment*, Vol. 9, No. 3, pp. 161-171.
- [16] Udo de Haes, H. A., Heijungs, R., Suh, S., Huppel, G., 2004, "Three Strategies to Overcome the Limitations of Life-Cycle Assessment," *Journal of Industrial Ecology*, Vol. 8, No. 3, pp. 19-32.
- [17] Frischknecht, R., et al., 2005, "The ecoinvent Database: Overview and Methodological Framework," *Int. J. Life Cycle Assessment*, Vol. 10, No. 1, pp. 3-9.
- [18] Lenzen, M., 2000, "Errors in Conventional and Input-Output Based Life Cycle Inventories," *Journal of Industrial Ecology*, Vol. 4, No. 4, pp. 127-148.
- [19] Field, F., Kirchain, R., Clark, J., 2001, "Life-Cycle Assessment and Temporal Distributions of Emissions: Developing a Fleet-Based Analysis," *Journal of Industrial Ecology*, Vol. 4, No. 2, pp. 71-91.
- [20] Huijbregts, M., 2002, *Uncertainty and Variability in Environmental Life-Cycle Assessment*, Ph.D. thesis, Department of Environmental Studies, University of Nijmegen, The Netherlands.
- [21] Maurice, B., Frischknecht, R., Coelho-Schwartz, V., Hungerbühler, K., 2000, "Uncertainty Analysis in Life Cycle Inventory: Application to the Production of Electricity with French Coal Power Plants," *Journal of Cleaner Production*, Vol. 8, No. 2, pp. 95-108.
- [22] Huijbregts, M. A., et al., 2001, "Framework for Modelling Data Uncertainty in Life Cycle Inventories," *The International Journal of Life Cycle Assessment*, Vol. 6, No. 3, pp. 127-132.
- [23] Tan, R. R., Culaba, A. B., Purvis, M. R. I., 2002, "Application of Possibility Theory in the Life-Cycle Inventory Assessment of Biofuels," *International Journal of Energy Research*, Vol. 26, No. 8, pp. 737-745.
- [24] Chevalier, J.-L., Téno, J.-F. L., 1996, "Life Cycle Analysis with Ill-Defined Data and its Application to Building Products," *The International Journal of Life-Cycle Assessment*, Vol. 1, No. 2, pp. 90-96.
- [25] Sonnemann, G. W., Schuhmacher, M., Castells, F., 2003, "Uncertainty Assessment by a Monte Carlo Simulation in a Life Cycle Inventory of Electricity Produced by a Waste Incinerator," *Journal of Cleaner Production*, Vol. 11, No. 3, pp. 279-292.
- [26] Björklund, A. E., 2002, "Survey of Approaches to Improve Reliability in LCA," *The International Journal of Life Cycle Assessment*, Vol. 7, No. 2, pp. 64-72.
- [27] Vigon, B. W., Jensen, A. A., 1995, "Life Cycle Assessment: Data Quality and Databases Practitioner Survey," *Journal of Cleaner Production*, Vol. 3, No. 3, pp. 135-141.
- [28] Weidema, B. P., Wesnaelg, M. S., 1996, "Data Quality Management for Life Cycle Inventories – An Example of Using Data Quality Indicators," *Journal of Cleaner Production*, Vol. 4, No. 3, pp. 167-174.
- [29] Weidema, B., 1998, "Application Typologies for Life Cycle Assessment," *The International Journal of Life Cycle Assessment*, Vol. 3, No. 4, pp. 237-240.
- [30] Finnveden, G., Lindfors, L.-G., 1998, "Data Quality of Life Cycle Inventory Data – Rules of Thumb," *The International Journal of Life Cycle Assessment*, Vol. 3, No. 2, pp. 65-66.
- [31] Weidema, B. P., 1998, "Multi-user Test of the Data Quality Matrix for Product Life Cycle Inventory Data," *The International Journal of Life Cycle Assessment*, Vol. 3, No. 5, pp. 259-265.
- [32] Suh, S., Huppel, G., 2002, "Missing Inventory Estimation Tool using Extended Input-Output Analysis," *The International Journal of Life Cycle Assessment*, Vol. 7, No. 3, pp. 134-140.
- [33] Fayyad U. et al. (1996), "From Data Mining to Knowledge Discovery in Databases", *AI Magazine* (Vol 17, No 3), 37-54
- [34] Tan, Steinbach and Kumar, *Introduction to Data Mining*, Addison Wesley, 2006.
- [35] Navarro, "A guided tour to approximate string matching," *ACM Computing Surveys* 33 (1): 31-88, 2001.
- [36] Lawson and Hanson, "Solving Least Squares Problems," *Classics in Applied Mathematics*, SIAM 1995.
- [37] N. Sundaravaradan, D. Patnaik, N. Ramakrishnan, M. Marwah, and A. Shah, *Discovering Life Cycle Assessment (LCA) Trees from Impact Factor Databases*, communicated to the Twenty-Fifth Conference on Artificial Intelligence (AAAI'11), Special Track on AI and Sustainability, to be held in San Francisco, CA, Aug 2011.