Constraint Driven Model Using Correlation and Collaborative Filtering for Sustainable Building

Hsin-Yu Ha[1], Shu-Ching Chen[1], Yimin Zhu[2], Steven Luis[1], Scott Graham[1], Shahin Vassigh[3]
[1]*School of Computing and Information Sciences*
{hha001, chens, luiss, grahams}@cs.fiu.edu
[2]*Department of Construction Management*
zhuy@fiu.edu
[3]*School of Architecture*
shahin.vassigh@fiu.edu
Florida International University
Miami, FL 33199, USA

Abstract

Sustainable building has emerged as an important topic due to the fact that it can significantly reduce the impact of buildings and their operation on the natural environment and more efficiently utilize resources throughout a building's life-cycle. When compared with a traditional building-design process, integrated project delivery has proven to be more efficient, and is thus gaining wider acceptance for many sustainable building projects. However, managing design and construction from different disciplines is still challenging. Conflicts among constraints are often not identified at the right design stage, which results in multiple iterations of the design process. In this paper, a novel constrain-driven model that enhances design processes through better management of constraints and thus delivers optimal design solutions with higher energy performance is proposed. Multiple Correspondence Analysis was applied to capture the correlations between different items (parameter-value pairs) and classes (constraints). Meanwhile, it integrated Collaborative Filtering methods and Constraint Satisfaction Problem to train and refine the proposed model. Finally, we have applied our model to a synthetic data sets to demonstrate its performance.

1. Introduction

In recent decades, there has been a growing interest in environmentally sustainable buildings use energy efficiently and reduce waste/pollution. To outperform traditional building-design processes, multidisciplinary integration has proven to be useful in the early design stage of sustainable building in terms of performance-based considerations [22]. As Integrated Project Delivery (IPD) [15] is used in more and more sustainable building projects, it is necessary to explore the entire scope of constraints from all the involved disciplines and capture their relations to each objective, in order to obtain valid solution sets. When facing constraints from multiple disciplines, it is critically important to identify conflicts among constraints at the early design phases to avoid multiple redesign processes. Our goal is to enhance the efficiency of the design process and to deliver optimal solution sets that achieve higher energy performance.

In this paper, a novel constraint-driven model is proposed to provide an efficient way to quickly generate preferred solution sets that the users, i.e. architects, engineers, etc., can use to effectively choose design options. Their selections are utilized to refine the model in terms of execution time and better prioritized solution sets. The proposed model aims to support dynamic evolution of specific constraints from different disciplines and to raise the awareness of the impact and interaction of each expert's design decisions. Identification and management of constraints' evaluation and interaction are barriers that currently have not been successfully resolved by most Building Information Modeling (BIM) tools. [5] A set of synthetic data was generated to demonstrate our model as a proof of concept.

The rest of the paper is organized as follows: In section 2, we give an overview of related works. Section 3 presents the overall framework of the proposed model along with an introduction of its three major components. An analysis of our experimental results is discussed in Section 4. Lastly, we conclude and discuss our future work in Section 5.

IEEE IRI 2012, August 8-10, 2012, Las Vegas, Nevada, USA

2. Related Work

One of the major components in our proposed model is known as constraint satisfaction, which is originally derived from mathematical problems and has proven to be one of the most successful problem solving paradigms in Artificial Intelligence. Basically, a constraint satisfaction problem is defined by a set of variables and a set of constraints. Each variable has its corresponding finite domain of possible values and each constraint exists among the variables. The goal of a Constraint Satisfaction Problem (CSP) is to assign values to each variable that satisfies all the constraints.

Since many real problems can be formulated as CSPs, there has been great research interest in applying CSPs to areas such as structural design [8], scheduling, planning, and resource allocation [16, 21, 7]. In [8], a fuzzy control was applied on the constraint satisfaction to resolve the structural design problem caused by statistical partitions and to further optimize design performance. In [16], CSPs were proposed to be deployed to enhance the outcomes of project scheduling and to overcome the inconsistency and resource conflicts. However, to the best of our knowledge in the sustainable building research, there have been no studies of leveraging CSPs to avoid the design conflicts that appear in traditional building design procedures. In addition, unlike [8, 16, 21, 7] which aim to generate a group of valid solution sets which satisfy all the defined constraints and optimize the results to better achieve a single objective, the proposed model aims to capture the relationships between constraints and multiple objectives, i.e. energy consumption, time, cost, etc. Users can thus better analyze results and redefine constraints.

Valued CSP [19] (as well as semi-ring CSP [3]) are widely used to associate weights (costs) with tuples by means of a so-called valuation structure to specify costs. [11, 23, 10] applied one specific subset of Valued CSP, called weighted CSP (WCSP), which assigns costs as natural numbers or infinite numbers and aims to find solution sets with minimum costs. Instead of directly weighting a tuple itself, the utilization of Multiple Correspondence Analysis (MCA) can specifically assign weights to each feature-value pair based on its contribution to the involved constraints, thus the summation of the included feature-value pair's weights could indicate each tuple's probability to satisfy the corresponding constraint. In addition, the Collaborative Filtering (CF) method was adapted to take user feedback as one of its training factors so the proposed model can take the opinions of domain experts into account. At the end, the combination of MCA and CF method's results will be leveraged to repeatedly train the proposed model. This iterative process could speed the design process by ruling out improper results.

3. Overall Framework

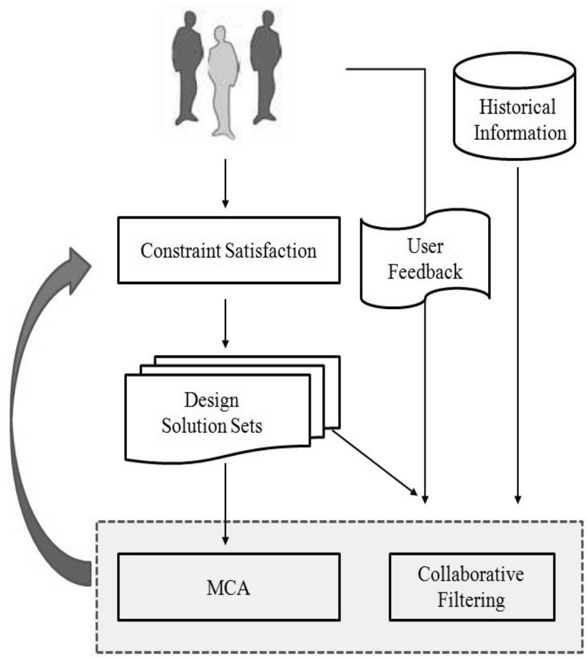

Figure 1: Framework

Figure 1 illustrates the overall framework of our model. At the beginning, users define a set of constraints which will be formulated as a CSP representation, including all involved design variables and the finite domain range from each discipline. Constraint satisfaction avoids the conflicts among those constraints and processes constraints to generate all valid design solution-sets. Once the ranked solution sets are produced, users can indicate their preferences for each set based on their domain knowledge. With the user's feedback and previous design records from historical information, a Collaborative Filtering method can be utilized to capture the user's interests and propose a better prioritized ranked list. Meanwhile, having all possible solution sets is considered ideal to adopt Multiple Correspondence Analysis to obtain the correspondence between items (variables-value pairs) and classes (constraints). In other words, a variable-value pair's contribution in satisfying each constraint could be identified. Finally, with the use of MCA and CF methods, the proposed model can be trained iteratively and keep on refining the ranked results.

3.1. Constraint Satisfaction Problem - DE + PW-AC

A Constraint Satisfaction problem is formally composed by a tuple *(X,D,C)*, whose components are defined below,

- X = { X_1,...,X_n}is a finite set of n variables.

- D = { $D(X_1)$,...,$D(X_n)$} represents a finite set of domains. For each $x_i \in$ X, $D(X_i)$ is the finite domain of the variable.

- C = { C_1,...,C_m}is a finite set of m constraints. Each constraint C_i is represented over an ordered subset of variables $var(C_i)$. The size of $var(C_i)$ is known as the *arity* of the constraint. Thus, a binary constraint has arity equal to 2 and non-binary constraint has arity greater than 2.

Given the above definition, we are motivated to adopt CSP in sustainable building design process to elude possible design disagreements and smoothly expedite development. Although CSP seems to fit into building design scenarios, it is necessary to transform non-binary constraints to binary constraints in order to accommodate problems the happen in reality and take advantage of the plentiful research already done for binary constraints.

The two most well-known transformations are the dual encoding (DE) [9] and the hidden variable encoding (HVE) [14]. In [18], dual encoding has demonstrated its capability of strong filtering especially when it was engaged in a low cost specialized algorithm, such as PieceWise Arc Consistency (PW-AC), which makes it more competitive against hidden variable encoding in terms of computation time. Accordingly, we decided to make use of DE's strength in conjunction with PW-AC to support search efficiency in the proposed model.

3.1.1 Dual Encoding

Dual encoding was first derived from the relational database community and was later brought to CSP research topics. Here, it is briefly introduced with the example shown in Figure 2. With the DE transformation, the constraints of the original formulation are converted into variables, like $\{V_{C1}, ..., V_{C4}\}$, which are also referred to Dual Variable, and the variables from the original CSP are referred to ordinary variables. The set of tuples which belong to one constraint appear as its domain in the new representation. Upon transformation, a binary constraint can be defined within two dual variables to indicate whether they shared the same ordinary variables and prevent inconsistent value initialization among the shared variables

3.1.2 PieceWise-Arc Consistency

In [9], PieceWise Arc Consistency was sketched as an Arc Consistency (AC) algorithm for dual encoding to ensure a CSP (X,D,C) is arc consistent, meaning there is no empty domain and each constraint C_i in *C*, and each variable $x_j \in$ vars(C_i), would be successfully assigned at least one valid value which satisfies the constraint C_i.

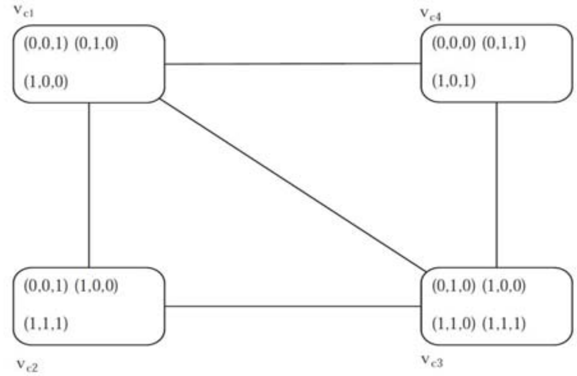

Figure 2: An example of using dual encoding transformation

PW-AC is as mainly separated into three components as presented in the pseudo code shown as Figure 3. First, the algorithm loops through all the variables to fetch the number of the valid tuples for each group by calling the function $GroupOf(S(V_i, V_j))$ (line 3-6). Next, the groups with zero supported tuples are propagated and checked to see whether there is any empty domain which results in $Inconsistency$ for the CSP. (line 7-19) Last, identified groups with zero supported tuples are used to revise the number of tuples for the corresponding group.

3.2. Multiple Correspondence Analysis

Multiple Correspondence Analysis is a descriptive/exploratory data analytic technique extended from the standard correspondence analysis (CA) to analyze not just simple two-way but multi-way tables for capturing correspondence within the rows and columns [17]. Enlightened by MCA's ability of indicating the correlation between columns and rows, we have investigated using it to analyze multimedia data instances represented by a set of low-level features that capture the correspondence between items (feature-value pairs) and classes (subjects). On top of this utilization, the correlation between feature-value pairs and classes was adopted as an indication of their similarity by calculating the inner product of each feature-value pair and class [12, 13].

To accommodate MCA in the proposed framework, assumptions were made as follows: Suppose CSP will come up with *I* valid solution sets together with the historical data (if any) and that each solution set is characterized by a set of designed variables. To better analyze quantitative data, each numeric variable should be discretized into several intervals so they can all be converted into nominal values. Given K nominal variables, each variable has J_k intervals (feature-value pairs), and the sum of all J_k is denoted as J which represents the total number of items. The indicator matrix

```
function PW − AC
    Q ← ∅
    initialize all group counters to 0
    for each variable v_i
        for each variable v_j constrained with v_i
            for each tuple τ ∈ D(v_i)
                counter(GroupOf(S(v_i, v_j), τ)) ← counter(GroupOf(S(v_i, v_j), τ)) + 1
    for each variable v_i
        for each variable v_j constrained with v_i
            for each group s_l(v_i, v_j)
                if counter(s_l(v_i, v_j)) = 0
                    put s_l(v_i, v_j) in Q
    return Propagation

function Propagation
    while Q is not empty
        pop group s_l(v_i, v_j) from Q
            δ ← ∅
            δ ← Revise(v_i, v_j, s_l(v_i, v_j))
                if D(v_j) is empty return INCONSISTENCY
                for each group s_l'(v_j, v_k) in δ put s_l'(v_j, v_k) in Q
    return CONSISTENCY

function Revise(v_i, v_j, s_l(v_i, v_j))
    for each tuple τ ∈ D(v_j) where τ ∈ sup(s_l(v_i, v_j))
        remove τ from D(v_j)
        for each group s_l'(v_j, v_k) that includes τ
            counter(s_l'(v_j, v_k)) ← counter(s_l'(v_j, v_k)) − 1
            if counter(s_l'(v_j, v_k)) = 0
                add s_l'(v_j, v_k) to δ
    return δ
```

Figure 3: PieceWise-Arc Consistency algorithm

is denoted by X with size $I \times J$. The inner product of X, i.e. Burt matrix Y is introduced with size XTX to reduce the computation time. Next, let the grand total of the Burt matrix be N with the probability matrix $Z = Y/N$. The vector of the column totals of Z is denoted as a mass matrix with size $1 \times J$, and D represents the diagonal matrix of M thus $D = diag(M)$. Through the singular value decomposition (SVD) presented in Equation (1), the principle components can be provided by MCA and the top two principle components will be further used to project data into a new space since over 95% of the total variance can be captured by the first two principal coordinates [1].

$$(D)^{1/2}(Z - MM^T)(D^T)^{-1/2} = P\Delta Q^T \tag{1}$$

Given two-dimensional principle components P and C, the correlation between different solution sets (also called feature-value pairs) and the constraints (also referred to classes) can be represented by angles calculated as shown in Equation (2).

$$Angle_m^i = arccos(\frac{P_m^i \cdot C}{|P_m^i||C|}) \tag{2}$$

The angles will be continually applied to the weight conversion as an indication of the similarity between each feature-value pair and constraint, as shown in Equation (3).

$$weight_m^i = \pm(1 + \cos(Angle_m^i \times \pi/180)) \tag{3}$$

Lastly, the overall evaluation regarding the relationship among each design solution and constraint will be calculated by summing up all the weights within a design solution as depicted in Equation (4). It is denoted as S_i for the ith solution set, where the total number of design variables is M and the total number of solution sets is denoted as N.

$$S_i = \sum_{m=1}^{M} weight_m^i, i \in \{1, 2, ..., N\} \tag{4}$$

The weight will be subsequently integrated with the weight produced by Collaborative Filtering methods.

3.3. Collaborative Filtering

The motivation of adopting a CF method is to observe a user's interests and take those into consideration while ordering the final ranked solution sets so users can quickly decide on the solution sets with higher energy performance. Specifically, CF method aims to predict the utility of solution sets to a particular user based on the feedback from the population of all users. Collaborative filtering algorithms can be generally categorized into two classes as follows: *Memory-based Algorithms* compute the similarity between users or items based on user's past rating data to make rating predictions. In contrast to the memory-based algorithms, *Model-based Algorithms* are applied to find patterns or correlation on training data using data mining machine learning algorithms by using them to predict the rating of *unseen items*. A hybrid model has manifested its ability to improve the prediction performance and overcome the limitations of single models, especially the cold start problem and sparsity [20]. Hence, we decide to amalgamate the use of MCA (considered as a memory-based algorithm) and a model-based algorithm to benefit from both models.

The *Pearson Correlation Coefficient* is selected to play the role of Model-based algorithm, not only because it is a well-known, widely used algorithm in CF domain, but also because it has proven to outperform Bayesian-clustering and vector similarity methods [6]. To demonstrate its applicability, suppose we obtain a user rating database that consists of a set of votes $v_{i,j}$ corresponding to the vote for user i on solution set j, and the set of solution sets on which user i has voted is denoted as I_i, then the mean vote for user i could be defined as:

$$\bar{v}_i = \frac{1}{|I_i|} \sum_{j \in I_i} v_{i,j} \tag{5}$$

The basis for the weights indicating the similarity between each user *i* and the active user which could be defined through applying Pearson correlation coefficient is:

$$w(a,i) = \frac{\sum_j (v_{a,j} - \bar{v}_a)(v_{i,j} - \bar{v}_i)}{\sqrt{\sum_j (v_{a,j} - \bar{v}_a)^2 \sum_j (v_{i,j} - \bar{v}_i)^2}} \tag{6}$$

where *a* represents all the active users who have voted in the rating system.

The predicted vote of the active users for solution set j could then be calculated based on similarity captured above:

$$Pa,j = \bar{v}_a + \kappa \sum_{i=1}^{n} w(a,i)(v_{i,j} - \bar{v}_i) \tag{7}$$

where n is the number of users without weight equal to zero. Note that the predicted vote calculated for each solution in Equation (7) is constraint-wise.

Finally, we are able to derive the weighting factor by combining the weights from both Equation (4) and Equation (7) as follows:

$$Weight_f\{j\} = \alpha \frac{S_j}{\sum\limits_{j=1}^{N} S_j} + \beta \frac{Pa,j}{\sum\limits_{j=1}^{N} Pa,j} \tag{8}$$

where $Weight_f$ represents the fused score, while the α and β are the tuning parameters which are set to be 0.5 and 0.5 in the experimental analysis.

4. Experimental process with hypothetical data

Without having actual sustainable building data, a set of synthetic data were created to demonstrate the proof-of-concept prototype. As shown on Figure 4, the representation of each constraint will be a formula and the domain of each variable will also be provided. Initially, all the possible solution sets which satisfy the defined constraints will be generated through the CSP process and treated as MCA's input. Thus the correlation between feature-value pair and constraint can be presented as an angle and further converted to a transaction weight. Then the model can display the solution set with higher transection weight and let users easily provide their feedback and decide their design option. The feedback will be stored as history records and used to re-display the results which reflect the user's priority.

Our contribution to sustainable building design provides a whole new approach that allows users to quickly select their preferred design options from a reduced domain without violating any constraints. Optimization methods were also proposed to enhance the design process; however, they cannot prevent the optimal solutions from violating critical constraints and might result in time wasted during re-design processes. Second, the proposed approach derives all the possible solution sets from the given variable's domain and the defined constraints, therefore we can not only ensure that solution sets satisfy all the constraints but might also discover solution sets with greater performance that were not used as design options in the past.

Figure 5 shows the user interface (UI), where users can upload their defined constraint file and provide their feedback toward MCA ranked solution sets for each constraint.

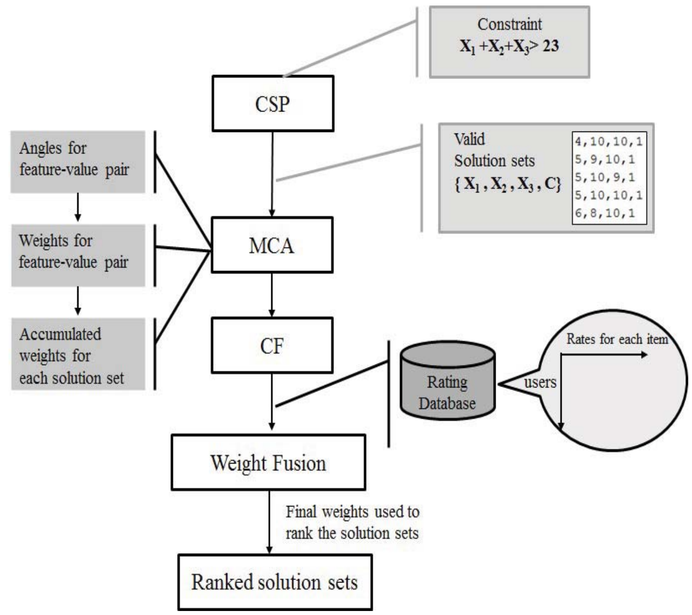

Figure 4: Process with synthetic data

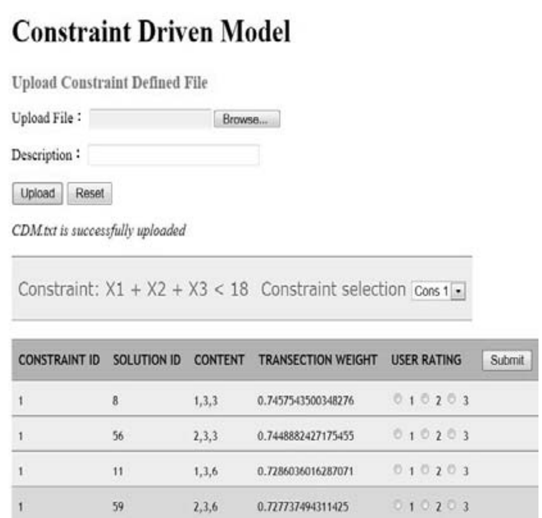

Figure 5: User interface

5. Conclusions and Future work

In this paper, a novel approach based on CSP, MCA, and Collaborative Filtering method is developed to enhance the design process of sustainable buildings by generating optimal solution sets with higher energy performance. With a set of well-defined constraints, we can demonstrate that the model is able to avoid the conflicts among these constraints, rank the solution sets based on correlation captured by MCA, and iteratively take all users' feedback into account to evolve over time. To cope with the actual design data, we will need to enhance the algorithm efficiency by further utilizing MCA's results in variable ordering or value ordering [2, 4]. In addition, we will introduce a case study with the major constraints as shown in Figure 6 to check whether the results generated by our model are competitive with the final design decision in the case study.

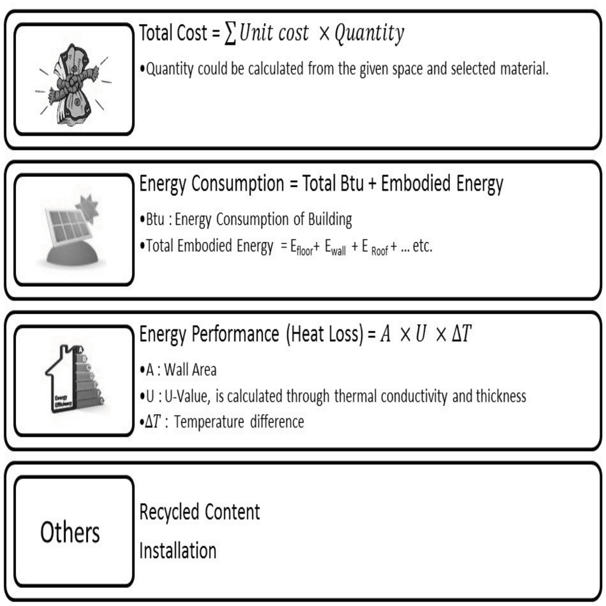

Figure 6: Major Constraints

6. Acknowledgement

This research was supported by the FIU Faculty Research Support Program (FRSP) and NSF award Nos. 0833093 and 1000136. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation and FIU.

References

[1] Chapman and Hall/CRC, 2006.

[2] P. Beck, J.C.and Prosser and R. Wallace. Toward understanding variable ordering heuristics for constraint satisfaction problems. *In: Proceedings of the Fourteenth Irish Artificial Intelligence and Cognitive Science Conference (AICS03)*, 2003.

[3] S. Bistarelli, U. Montanari, F. Rossi, T. Schiex, G. Verfaillie, and H. Fargier. Semiring-based csps and valued csps: Frameworks, properties, and comparison. 4(3):199–240, 1999.

[4] S. Bittle and M. Fox. Learning and using hyper-heuristics for variable and value ordering in constraint satisfaction problems. *In Proceedings of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference*, pages 2209–2212, 2009.

[5] S. Boddy, Y. Rezgui, G. Cooper, and M. Wetherill. Computer integrated construction: A review and proposals for future direction. *Advances in Engineering Software*, 38(10):677–687, 2007.

[6] J. Breese, D. Heckerman, and C. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. *In Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, 1998.

[7] L. Cucu-Grosjean and O. Buffet. Global multiprocessor real-time scheduling as a constraint satisfaction problem. pages 42–49, 2009.

[8] Q. Guan and G. Friedrich. Fuzzy control over constraint satisfaction problem solving in structural design. *Fuzzy Systems, 1993., Second IEEE International Conference*, 2:1316–1320, 1993.

[9] P. Karagiannis, N. Samaras, and K. Stergiou. Arc consistency in the dual encoding of non-binary csps. *1st Workshop in Constraint Propagation and Implementation*.

[10] T. Kumar. Lifting techniques for weighted constraint satisfaction problems. *Tenth International Symposium on Artificial Intelligence and Mathematics*.

[11] J. H. M. Lee, T. W. K. Mak, and J. Yip. Weighted constraint satisfaction problems with min-max quantifiers. pages 769–776, 2011.

[12] L. Lin and M. L. Shyu. Weighted association rule mining for video semantic detection. 1:37–54, 2010.

[13] L. Lin, M. L. Shyu, G. Ravitz, and S. C. Chen. Video semantic concept detection via associative classification. pages 418–421, 2009.

[14] N. Mamoulis and S. K. Solving non-binary csps using the hidden variable encoding. 2239:168–182, 2001.

[15] K. Molenaar, N. Sobin, D. Gransberg, T. McCuen, S. Korkmaz, and S. Horman. Sustainable high performance projects and projects delivery methods, a state-of-practice report. Technical report, Charles Pankow Foundation and the Design-Build Institute of America, September 2009.

[16] M. A. Rigi and S. Mohammadi. Finding a hybrid genetic algorithm-constraint satisfaction problem based solution for resource constrained project scheduling. *International Conference on Emerging Technologies*, 2009.

[17] N. Salkind. SAGE Publication, Inc, 2007.

[18] N. Samaras and K. Stergiou. Binary encodings of non-binary constraint satisfaction problems: Algorithms and experimental results. 24:641–684, 2005.

[19] T. Schiex, H. Fargier, and G. Verfaille. Valued constraint satisfaction problems: Hard and easy problems. 1995.
[20] X. Su and T. M. Khoshgoftaar. A survey of collaborative filtering techniques. *Adv. in Artif. Intell.*, pages 4:2–4:2, 2009.
[21] B. Sun, W. Wang, and Q. Qi. Satellites scheduling algorithm based on dynamic constraint satisfaction problem. pages 167–170, 2008.
[22] P. Tavares and A. Martins. Energy efficient building design using sensitivity analysis –a case study. *Energy and Buildings*, 39(1):23–31, 2007.
[23] M. Zytnicki, C. Gaspin, and T. Schiex. A new local consistency for weighted csp dedicated to long domains. pages 394–398, 2006.