Aggregating Opinions to Design Energy-Efficient Buildings

**Leandro Soriano Marcolino[1], Boian Kolev[2], Samori Price[2],
Sreerag Palangat Veetil[3], David Gerber[3], Josef Musil[3], Milind Tambe[1]**
[1]Computer Science Department, University of Southern California, Los Angeles, CA, USA
[2]Computer Science Department, California State University, Dominguez Hills, Carson, CA, USA
[3]School of Architecture, University of Southern California, Los Angeles, CA, USA
sorianom@usc.edu, {bkolev1, sprice25}@toromail.csudh.edu,
{palangat, dgerber, jmusil}@usc.edu, tambe@usc.edu

Abstract

In this research-in-progress paper we present a new real world domain for studying the aggregation of different opinions: early stage architectural design of buildings. This is an important real world application, not only because building design and construction is one of the world's largest industries measured by global expenditures, but also because the early stage design decision making has a significant impact on the energy consumption of buildings. We present a mapping between the domain of architecture and engineering research and that of the agent models present in the literature. We study the importance of forming diverse teams when aggregating the opinions of different agents for architectural design, and also the effect of having agents optimizing for different factors of a multi-objective optimization design problem. We find that a diverse team of agents is able to provide a higher number of top ranked solutions for the early stage designer to choose from. Finally, we present the next steps for a deeper exploration of our questions.

Introduction

Aggregation of opinions is an exciting area of research, but the field still lacks sufficient concrete practical applications. It was recently shown (Marcolino, Jiang, and Tambe 2013; Marcolino et al. 2014) that a team of agents voting together is able to achieve good results in the Computer Go domain. However, it is still necessary to show the importance of social choice in more concrete real world problems.

We are now performing an interdisciplinary research, to explore a new domain to study a diverse team of voting agents: the design, architecture and engineering of buildings. Architectural design, Engineering and Construction (the AEC industry) constitutes the largest industry by total aggregate expenditure for the global economy. Given the scenario of global depletion or scarcity of resources, it is becoming more essential to adopt building design and AEC practices that optimizes for different performance criterias simultaneously, including energy efficiency, construction cost, social well-being, and life cycle cost, for example. Designing energy efficient buildings, in particular, is a major concern. With the continuous decrease of the availability of fossil fuels, and increasing environmental concerns, it is fundamental to seek building designs that have efficient energy usage while maintaining optimalities across the other competing factors intrinsic in all AEC design projects.

In this research-in-progress paper we demonstrate a diverse team of agents that combine their opinions in order to optimize early stage design decision making in architectural design. We show that a diverse team of agents is able to find a larger number of top-ranked solutions than each single agent. This provides the designer or design team greater flexibility into choosing a high-quality model.

We also explore the idea of creating diversity by making each agent optimize for a different factor of the building design problem (which is a multi-objective optimization problem). This single objective optimization is key to break a complex problem of building design into partial design problems, that are easier to frame and define. It also allows the reuse of existing software that optimizes for single objective problems. Moreover, this approach is closer to existing real world distribution of different professions within building design teams. It is, therefore, important to study how to combine the opinions of such agents.

We evaluate voting as a way to aggregate the opinions of these different agents. Traditionally, there are two different views of voting in social choice: voting can be seen as a way to aggregate the preferences of different individuals, or as a way to estimate the best possible decision. In this work we are combining the two views, by exploring if aggregating the opinions of agents that optimize for their own preferences (factors) leads to solutions that are close estimates of the best possible one.

Hence, this work presents the following contributions: (1) we show a new, concrete practical application for aggregation of opinions; (2) we map and test the ideas in Marcolino, Jiang, and Tambe (2013) in a completely new problem domain; (3) we explore the performance of a team of agents that optimize for their own preferences, instead of optimizing for the whole multi-objective function. Finally, we discuss our results and show next steps for exploring deeper the questions raised by this work.

Related Work

Preference handling is an important area of research. Diversity was shown to be an important concept when teams of agents collaborate by sharing their opinions. LiCalzi and Surucu (2012) and Hong and Page (2004) focus on modeling human decision makers, and propose models where the agents are able to know the utility of the solutions that they find, and the team converges to the best solution found by one of its members. In complex problems such information is not available, and the team must resort to other coordination mechanisms, such as voting, to take a common decision. Lamberson and Page (2012) study diversity in the context of forecasts, where the solutions are real numbers and the team can take the average of the opinion of its members. Their model does not capture domains where there are a discrete number of possible actions.

Marcolino, Jiang, and Tambe (2013) and Marcolino et al. (2014) study teams of agents that vote together for solving complex problems. They present empirical results in the Computer Go domain. Although their models are not only limited to Computer Go, it is still an open question if such models would apply to different domains, especially real world applications. Moreover, in their work the idea of agents with different preferences (optimizing for different factors) was not explored.

Many recent works in social choice present new models and theories (Caragiannis, Procaccia, and Shah 2013; Soufiani, Parkes, and Xia 2012). However, the field still lacks more practical applications beyond elections. Mao, Procaccia, and Chen (2013) studies the performance of combining the opinion of human subjects into solving problems, by using different voting rules. However, the problems imposed to the humans are still not yet real world problems. They use a sliding squares 8-puzzle game, and they also ask the subjects to count the number of dots in different pictures. In this paper we show the applicability of social choice in a very important real world domain: the architectural design domain.

Architectural design processes are highly complex and multivalent; including many large-scale problems, that contain a great number of parameters and/or highly complex couplings of parameters (Flager, Gerber, and Kallman 2014). Traditionally, designers are limited by time, and can evaluate only a small number of possible solutions (Gerber and Lin 2013). While almost all specific objectives of design are different in its mathematical formulation, these are considered in parallel within the conjunctive capacity of individual collaborative professionals.

Various multi-objective optimization approaches have been introduced to tackle this problem, that are called multidisciplinary design optimization (MDO) (Lin and Gerber 2012). One such approach is presented in Sisk, Miles, and Moore (2003), that uses a Genetic Algorithm (GA) to aid in the design process. Another GA based approach in the design domain is the Pareto Genetic Algorithm-Based Collaborative Optimization in Building Design (Ren et al. 2011). They present a pilot study to find out possible feasible design solutions for an office space by minimizing weight of the column and minimizing heating and cooling load. The GA approach is compared with an agent based approach, where agents negotiate between different possible solutions. Another interesting application of multi-objective GAs in Architectural Design domain can be found in the work of EZCT Architecture and Design Research on Seroussi Pavilion (Feringa 2008). EZCT evolves a large architectural volume encoded with only a few points, by using voronoi diagrams and an evaluation function based on light radiance and pedestrian circulation. Their work uses the multi-objective evolutionary method developed by M. Schoenauer for Topological Optimum Design (Hamda, Roudenko, and Schoenauer 2002).

Methodology

Design Domain

In early stage of design, the design team explores a variety of forms that could be suitable for the final functioning building. A broad range of possible solutions are intuitively analyzed for optimal solutions. Typically energy performance assessments are made after this initial design phase, where the analysis is performed on a very limited set of design alternatives rather than to earlier stage design decisions (Radford and Gero 1980). Currently there is limited direct and validated feedback between the domains of design and energy simulation available during the early stages of the design process. However, it has been acknowledged that such feedback has the highest potential impact on the overall building performance (Bogensttter 2000). Many preparatory building information modeling (BIM) tools are already developed with their own energy analysis and simulation packages. It was also found that the automation and integration of energy performance analysis into early stage design will contribute to higher performing buildings, especially when considered in the early stage design decision making (Lin and Gerber 2012).

In this work we use the H.D.S Beagle system (Gerber and Lin 2013). Beagle is a MDO software framework that assists users in the early stage design of buildings. It incorporates an optimization methodology that combines parametric modeling with multi-objective optimization through an integrated platform for enabling rapid iteration and trade-off analysis across the domains of design, energy use intensity, and finance.

First, the designer uses Autodesk Revit to create a parametric model. This is a base model of the building, containing a set of parameters that can be modified within a specified range, allowing the creation of many possible variations of the base model. These parameters and their valid range are also defined by the designer. A simple example can be seen in Figure 1, where the parameters $X1$ and $Y1$ are being used to specify the position of the lower left corner of the building relative to the site boundary.

Therefore, by modifying these parameters, it is possible to create many different building design variants. Beagle uses a Genetic Algorithm (GA), in order to optimize the building design based on three objectives. Each solution is analyzed in the multi-objective optimization framework, according to the following three factors: $(S_{obj}, E_{obj}, F_{obj})$. The objective functions are: $S_{obj} : \max SPCS$; $E_{obj} : \min EUI$; $F_{obj} : \max NPV$. SPCS is the Spatial Programming Compliance Score, EUI is the Energy Use Intensity and, finally, NPV is

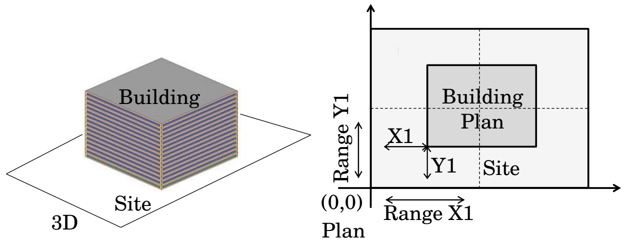

Figure 1: A parametric model of a building, showing two parameters: $X1$ and $Y1$.

the Net Present Value, defined as follows.

SPCS defines how well a building conforms to the project requirements of a user. It measures how close the area dedicated to different activities in a building is to a given specification. Let $\mathbf{L}$ be a list of activities (in our models, **L**=<Office, Hotel, Retail, Parking>), $area(l)$ be the total area in a building dedicated to activity l and $requirement(l)$ be the area for activity l given in a project specification. SPCS is defined as:

$$SPCS = 100 * \left(1 - \frac{\sum_{l \in \mathbf{L}} |area(l) - requirement(l)|}{|\mathbf{L}|}\right)$$

EUI regulates the overall energy performance of the building. This is an estimated overall building energy consumption in relation to the overall building floor area. The process to obtain the energy analysis result is automated in Beagle through a DOE-2.2 simulation engine, implemented through Autodesk Green Building Studio (GBS) web service (`https://gbs.autodesk.com/GBS/`).

Finally, NPV measures the financial performance for the whole building life cycle, given by:

$$NPV = \left(\sum_{t=1}^{T} \frac{c_t}{(1+r)^t}\right) - c_0,$$

where T is the Cash Flow Time Span, r is the Annual Rate of Return, c_0 is the construction cost, and c_t = Revenue − Operation Cost.

In the end of the optimization process, the GA Pareto-ranks all the solutions. Generally, there will be a set of first-ranked solutions, and a designer must choose one among them according to her own subjective qualitative and quantitative evaluation. Many options can affect the execution of the GA, including: initial population size, size of the population, selection size, crossover ratio, mutation ratio, maximum iteration, and material systems. More information about the Beagle system can be found at Lin and Gerber (2012).

In this work we modified the original Beagle system. First, we discretize the set of possible values for each parameter. This approach is possible because Beagle is an early stage design decision making tool, so it is not necessary to have a high degree of numeric floating point precision. Hence, we also modified the mutation of the GA algorithm. In our system, each time a parameter is selected for mutation, it is replaced by randomly selecting one of the possible discretized values. This makes it possible to explore voting approaches in the design domain, as there is a discretized set of possible values for each parameter.

We also modified the procedure that generates the initial population. In the original system, the initial population is always the same across different runs (given the same set of options for the GA). We randomized the initial population procedure, in order to improve the test of the hypothesis of the value of diverse teams over uniform teams made of copies of a single agent. For such a test, it is necessary that each copy runs a stochastic search procedure, so that they do not end up all picking the same solutions. Of course, the crossover and mutation also occurs probabilistically in the system.

Furthermore, we modified the system to make it possible for the user to run the optimization according to only one of the factors of the multi-objective optimization. This modification was executed in order to make each GA have different preferences, so that we can explore what happens when we combine the solution of these different GAs, as we will show in the following Experiments section.

Agent Model

This paper is based on combining the research in the multidisciplinary design optimization and design computation domain with the team formation work presented in Marcolino, Jiang, and Tambe (2013), where a team of diverse agents is shown to outperform a uniform team in the Computer Go domain. Here we quickly review the model presented by Marcolino et al., and show how it maps to the design domain.

First, let us review the model. We assume there is a set $\mathbf{\Phi}$ of agents ϕ_i voting to decide an action a in a set of possible actions $\mathbf{A}$. The voting procedure happens at each world state ω_j in a set of world states $\mathbf{\Omega}$. We assume that we can rank the actions from best to worst and $\mathbf{U_j}$ is the vector of utilities of the actions in world state ω_j. The agents do not know the ranking of the actions, and will vote according to some decision procedure, characterized by a probability distribution function (pdf) over action ranks. Hence, each agent ϕ_i has a pdf $\mathbf{V_{i,j}}$ for deciding which action to vote for in state ω_j. Agents that have the same $\mathbf{V_{i,j}}$ in all world states will be referred as copies of the same agent.

Under this model, we define diversity as how different are the probability distributions of agents in $\mathbf{\Phi}$ in the set of world states $\mathbf{\Omega}$: $d = \frac{1}{|\mathbf{\Phi}|^2} \sum_{\omega_j \in \mathbf{\Omega}} \sum_{\phi_i \in \mathbf{\Phi}} \sum_{\phi_k \in \mathbf{\Phi}} \alpha_j H(\mathbf{V_{i,j}}, \mathbf{V_{k,j}})$, where H is a distance measure between two pdfs. Marcolino, Jiang, and Tambe (2013) uses the Hellinger Distance (Hellinger 1909), given by: $H(\mathbf{V_{i,j}}, \mathbf{V_{k,j}}) = \frac{1}{\sqrt{2}} \sqrt{\sum_{a \in \mathbf{A}} (\sqrt{V_{i,j}(a)} - \sqrt{V_{k,j}(a)})^2}$.

At each iteration, each agent will examine the current world state and submit its (single) opinion about which one should be the next action. The opinions are then combined using plurality voting, that picks as a winner the option that received the highest number of votes. We assume that ties are broken randomly.

Mapping

Now we are going to present a mapping of the design domain in the proposed agent model. As in any complex problem, the mapping is not perfect. In Marcolino, Jiang, and Tambe (2013), as well, the model does not capture all the complexities of the Computer Go domain.

In the design domain, we model each GA as an agent ϕ_i. The system allows for different parametrizations, so that different agents will tend to pick different options. Upon loading a parametric model, each GA outputs a solution (i.e., its preference).

As mentioned before, the final solution of each GA is given by a set of parameters. We model each parameter j as a world state ω_j. Moreover, the set of possible values for a given parameter is modeled as the set of possible actions $\mathbf{A}$, and the value assigned by a GA to one parameter is a certain action a of that GA in that world state.

Hence, given the outputs of all GAs of $\mathbf{\Phi}$, we combine their solutions by applying a voting rule across each parameter ω_j in the set of all possible world states (parameters) $\mathbf{\Omega}$. This leads to a new solution, that will be called as the solution of the team. In this paper we also show results using the mean and the median of the values proposed by each GA.

Note that in the agent model we have an utility U_j for each world state ω_j. However, we are really interested in the overall utility of a solution. For example, in Computer Go, the most important point is to win or lose the entire game. Similarly, in the design domain, we could assume an utility U_j for each parameter, and the parameter value that would lead to the optimal overall solution would have the highest U_j. Of course a wrong choice for one parameter could completely destroy the value of the final solution. This situation is actually similar to the Computer Go domain, where even though we might play perfectly in all board states, a mistake in one of them can ruin the entire game. If we always choose the optimal action for all world states, however, we will always reach the optimal solution.

A point of differentiation, however, is that the design domain is formulated as a multi-objective optimization problem. Therefore, we can actually have multiple optimal solutions, all tied in a Pareto-frontier. Such a situation is not captured by our agent model.

Hence, instead of picking only one solution from each agent of a team, and producing one single final solution, we can also generate a set of solutions for a team. As the agents have multiple top-ranked solutions, we can pick different combinations of top-ranked solutions from each agent. For each combination, we aggregate the selected solutions (by plurality voting, or by calculating the mean or the median of the proposed values for a parameter), producing one solution for the team. Note that one combination includes only one solution from each agent, but different combinations will select different top-ranked solutions from each agent.

Marcolino, Jiang, and Tambe (2013) shows that a team of diverse agents can outperform a uniform team if at least one agent plays better in at least one world state. Similarly, in the design domain, a diverse team of GAs should perform better than a uniform team of GAs if the different parametrizations lead to some agents having a higher probability of outputting the correct value for a certain parameter than the agents of the uniform team. In our experiments we will compare a diverse team of GAs and a uniform team composed by copies of the best GA.

Experiments

We run experiments using the Beagle system. In all experiments we use as input a parametric model similar to the one presented in Figure 1. We use 8 input parameters, specifying the position of each corner of the building. These parameters can vary within a range of 105 feet.

Experiment 1

First we study the effects of having each agent optimizing for a different factor. Hence, in our first experiment, we study a team of 3 agents: one optimizes for EUI, another for SPCS, and the last one for NPV. The solution of the team is computed using three different methods: mean, median and plurality voting. These methods are applied across each parameter. We compare the solutions of this team against the one of a GA that optimizes for all factors simultaneously (GA3). In this experiment we used the following options for the GA: Initial population size = 10, Size of the population = 15, Selection size = 10, Crossover ratio = 0.6, Mutation ratio = 0.1, Maximum iteration = 5.

We can see the result in Figure 2, where we plot all solutions found by GA3 and the solutions of the team according to the three different aggregation methods. The solutions obtained by mean and plurality voting were ranked as top 1 solutions, together with the best solutions found by GA3. This shows that we can obtain high-quality solutions by combining the solutions of different agents.

However, we also found that the top ranked solution of each individual agent were still ranked as top solutions when compared against the ones found by GA3. Therefore, it is still an open question if aggregating the opinion of agents with different preferences (optimizing for different factors) can lead to better solutions than multi-objective optimization approaches. As a next step, it is necessary to test more complex architectural models.

Note that in this case, each agent has only one top ranked solution, because they are optimizing for only one factor. Therefore, for each aggregation method the team also has only one solution. In the next experiment we will study the case where the agents have multiple top-ranked solutions.

Experiment 2

Now we study the effect of obtaining diversity by changing the options of the GAs. Here all agents optimize for all 3 factors of the multi-objective problem. We generate a team of 4 agents (Diverse), as defined in Table 1.

We ranked together the solutions proposed by all the agents, in order to identify the strongest one. As can be seen in Figure 3, Agent 3 had the highest number of top ranked solutions. Therefore, we also built a team composed by four copies of Agent 3 (Uniform).

We combine the solutions of the agents, by aggregating all the combinations of the three top ranked solutions. We

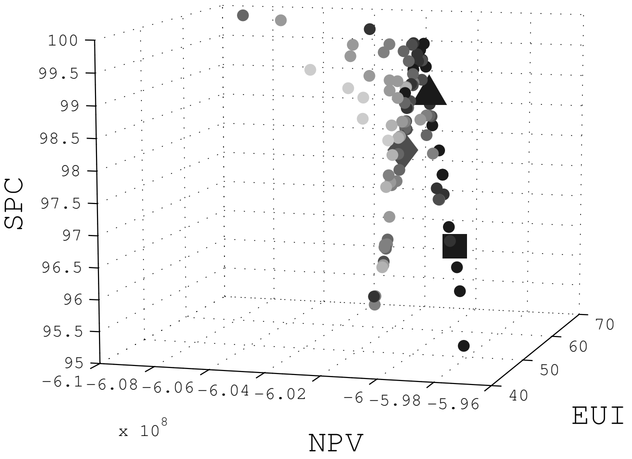

Figure 2: Value of the solutions. △: plurality voting; □: mean; ⋄: median; o: GA3 solutions. Dark colors have higher rank.

Agent	PZ	SZ	CR	MR
Agent 1	12	10	0.8	0.1
Agent 2	18	8	0.6	0.2
Agent 3	24	16	0.55	0.15
Agent 4	30	20	0.4	0.25

Table 1: GA parameters for the diverse team. Initial Population and Maximum Iteration were kept as constants: 10 and 5, respectively. PZ = Population Size, SZ = Selection Size, CR = Crossover Ratio, MR = Mutation Ratio.

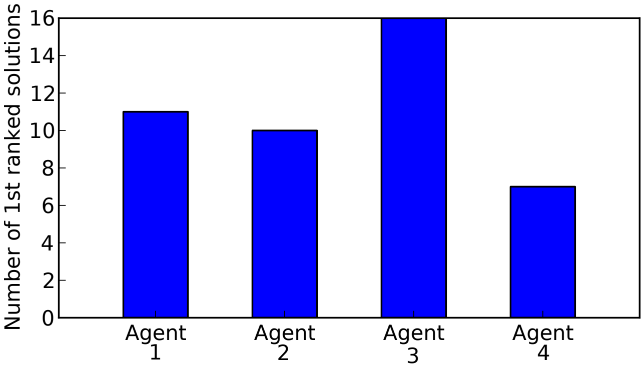

Figure 3: Number of 1st ranked solutions, when analyzing only the individual agents.

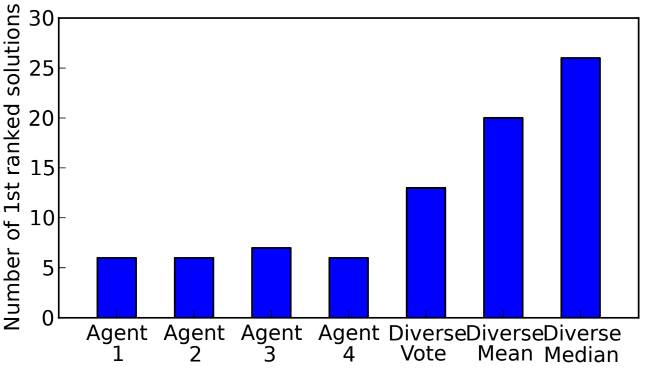

Figure 4: Number of 1st ranked solutions, for each agent and the diverse team.

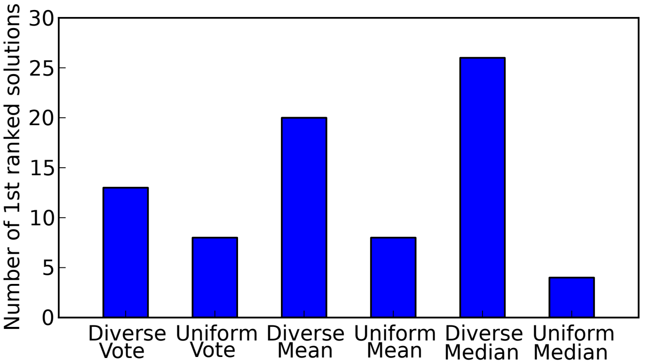

Figure 5: Number of 1st ranked solutions, for the diverse and the uniform team.

rank together the solutions proposed by all agents, and all solutions generated by the teams. In Figure 4 we show the number of unique first ranked solutions for each agent and the diverse team [1]. As can be seen, the diverse team was able to find a higher number of first ranked solutions than each individual agent, for all three aggregation methods. Median, in particular, was able to create 26 top ranked solutions, while each individual agent had only around 6 top ranked solutions.

In Figure 5, we show the number of unique first ranked solutions for the diverse and the uniform teams. As can be seen, for all aggregation methods, the diverse team found a higher number of top ranked solutions than the uniform team.

One could argue that we were able to find a higher number of unique top ranked solutions than each individual agent just because we are generating a higher number of solutions to be evaluated for the teams. However, during the GA process, each agent is evaluating 72 possible solutions. For each aggregation method, we are generating 81 solutions. Hence, we are testing only 9 solutions more for each team. The difference between the number of first ranked solutions of the teams and each individual agent is higher than 9 for the diverse team. Therefore, aggregating the opinions of these agents is really creating a large number of new top ranked solutions for the designer.

Next Steps and Discussion

As we saw, a diverse team was able to propose a high number of top ranked solutions, outperforming the number of top ranked solutions found by each individual agent and by the uniform team. Hence, we are able to facilitate a better design

[1] Here the number of top ranked solutions of each agent changes, because now we are ranking with a larger solution pool.

decision making in the early stage of design by providing the designer with a larger solution pool of high-quality solutions concerning energy and cost efficient designs. Given this larger pool, there is a higher likelihood that the designer will be able to find the best model according to her subjective evaluation.

As an immediate next step, we must conduct more experiments with complex parametric models and test whether our approach is advantageous in improving the solution quality and helping develop more complex design forms. In particular, our question about the performance of aggregating the opinions of agents that optimize for single factors is not yet completely answered, as in the parametric model explored in our experiments each agent was also able to find top ranked solutions (considering the multi-objective optimization problem). However, it is a good sign that aggregating their opinions led to top ranked solutions, and testing more complex models, where the agents will not be able to find top ranked solutions, will help clarify this issue. Moreover, even though the solutions found do not overcome the ones found by the multi-objective GA, they are tied in the top of the rank. Clearly, such an agent-based approach is easier to implement, and to run in parallel, enabling the designer to obtain results faster, leading to a quicker decision making process in the early stage design.

Moreover, we showed the potential of aggregating opinions in this domain. In particular, the early stage design domain accommodates a discretized set of possible solutions, allowing the exploration of voting. We saw that a team of agents using plurality voting was able to find top-ranked solutions for early stage design. Besides, the multi-objective optimization approach generates a rank of solutions. Hence, it also allows the exploration of other voting rules besides plurality, such as Borda or Copeland. Therefore, exploring ranked voting approaches is also an immediate next step. That would show even more the applicability of social choice and aggregation of opinions into real world, concrete problems.

Acknowledgments: This research is supported by MURI grant W911NF-11-1-0332, and the National Science Foundation under grant 1231001. 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 funding agencies. The authors would like to thank Shih-Hsin Eve Lin for her help with the Beagle system, and Albert Xin Jiang for useful discussions.

References

Bogensttter, U. 2000. Prediction and optimization of life-cycle costs in early design. *Building, Research & Information* 28:376–386.

Caragiannis, I.; Procaccia, A. D.; and Shah, N. 2013. When do noisy votes reveal the truth? In *Proceedings of the fourteenth ACM conference on Electronic commerce*, EC '13, 143–160. New York, NY, USA: ACM.

Feringa, J. 2008. Notes on the Potential of Simulation for Architectural Conception. In *Proceedings of the 28th Annual Conference of the Association for Computer Aided Design in Architecture*, ACADIA'08, 264–269.

Flager, F.; Gerber, D. J.; and Kallman, B. 2014. Measuring the impact of scale and coupling on solution quality for building design problems. *Design Studies* 35(2):180 – 199.

Gerber, D. J., and Lin, S.-H. E. 2013. Designing in complexity: Simulation, integration, and multidisciplinary design optimization for architecture. *Simulation* (April).

Hamda, H.; Roudenko, O.; and Schoenauer, M. 2002. Application of a Multi-Objective Evolutionary Algorithm to Topological Optimum Design. *Adaptive Computing in Design and Manufacture*.

Hellinger, E. 1909. Neue begründung der theorie quadratischer formen von unendlichvielen veränderlichen. *Journal für die reine und angewandte Mathematik* 136:210–271.

Hong, L., and Page, S. E. 2004. Groups of diverse problem solvers can outperform groups of high-ability problem solvers. *Proceedings of the National Academy of Sciences of the United States of America* 101(46):16385–16389.

Lamberson, P. J., and Page, S. E. 2012. Optimal forecasting groups. *Management Science* 58(4):805–810.

LiCalzi, M., and Surucu, O. 2012. The power of diversity over large solution spaces. *Management Science* 58(7):1408–1421.

Lin, S.-h. E., and Gerber, D. J. 2012. Designing-In Performance : A Framework for Evolutionary Energy Performance Feedback in Early Stage Design. *Automation and Construction* 38:59–73.

Mao, A.; Procaccia, A. D.; and Chen, Y. 2013. Better Human Computation Through Principled Voting. In *Proceedings of the 27th Conference on Artificial Intelligence*, AAAI'13.

Marcolino, L. S.; Xu, H.; Jiang, A. X.; Tambe, M.; and Bowring, E. 2014. Give a hard problem to a diverse team: Exploring large action spaces. In *Proceedings of of the 28th Conference on Artificial Intelligence*, AAAI'14.

Marcolino, L. S.; Jiang, A. X.; and Tambe, M. 2013. Multi-agent team formation: Diversity beats strength? In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence, IJCAI'13*, 279–285.

Radford, A., and Gero, J. 1980. Tradeoff diagrams for the integrated design of the physical environment in buildings. *Building and Environment* 15:3–15.

Ren, Z.; Yang, F.; Bouchlaghem, N. M.; and Anumba, C. J. 2011. Multi-disciplinary collaborative building design - A comparative study between multi-agent systems and multi-disciplinary optimisation approaches. *Automation in Construction* 20(5):537–549.

Sisk, G. M.; Miles, J. C.; and Moore, C. J. 2003. Designer Centered Development of GA-Based DSS for Conceptual Design of Buildings. (July):159–166.

Soufiani, H. A.; Parkes, D. C.; and Xia, L. 2012. Random utility theory for social choice. In Bartlett, P. L.; Pereira, F. C. N.; Burges, C. J. C.; Bottou, L.; and Weinberger, K. Q., eds., *Proceedings of the Neural Information Processing Systems Conference, NIPS'12*, 126–134.