Agents Vote for the Environment: Designing Energy-Efficient Architecture

Leandro Soriano Marcolino[1], David Gerber[34], Boian Kolev[2], Samori Price[2], Evangelos Pantazis [34], Ye Tian[1], 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
[4]Department of Civil Engineering, University of Southern California, Los Angeles, CA, USA
{sorianom,dgerber}@usc.edu, {bkolev1,sprice25}@toromail.csudh.edu, {epantazi,yetian,tambe}@usc.edu

Abstract

Saving energy is a major concern. Hence, it is fundamental to design and construct buildings that are energy-efficient. It is known that the early stage of architectural design has a significant impact on this matter. However, it is complex to create designs that are optimally energy efficient, and at the same time balance other essential criterias such as economics, space, and safety. One state-of-the art approach is to create parametric designs, and use a genetic algorithm to optimize across different objectives. We further improve this method, by aggregating the solutions of multiple agents. We evaluate diverse teams, composed by different agents; and uniform teams, composed by multiple copies of a single agent. We test our approach across three design cases of increasing complexity, and show that the diverse team provides a significantly larger percentage of optimal solutions than single agents.

Introduction

Sustainability and energy efficiency are major topics of inquiry. Given the current scenario of scarcity of resources, it is fundamental to find energy efficient solutions for the built environment globally. Designing energy efficient buildings, in particular, is an important area for concern as built infrastructures are the leading consumers of global energy. Once a building is constructed, it cannot be easily modified; and the building design has a major impact in the consumption of energy throughout its whole lifespan (Lin and Gerber 2014).

In architecture, human designers construct a schematic design of the building, using Building Information Modeling (BIM) software. Then, the designers can predict the energy efficiency of the building by using Building Performance Simulation tools. However, the variety of different designs a human can explore while seeking optimality is highly limited (most often given strict time constraints, besides the cognitive limits of the human designer (Flager, Gerber, and Kallman 2014)). A further challenge is that we cannot simply optimize for energy efficiency; as other factors, such as construction cost and design requirements, are also important to take into consideration. Hence, automated methods have the potential to greatly improve the designs.

Currently, Genetic Algorithms (GA) have been widely applied to better explore the solution space. For example, Beagle (Gerber and Lin 2013) is an example of the state of the art, where a human inputs a first design of the building, and the system generates a variety of high-quality variations in the pareto frontier. A human designer, then, can choose one of these variations, according to his/her own subjective evaluation, but most importantly with improved empirical certainty across the multiple objectives.

In this work, we further improve the state of the art, by aggregating the opinions of multiple agents. Although the benefits of combining opinions is known (Mao, Procaccia, and Chen 2013; Caragiannis, Procaccia, and Shah 2013), we present for the first time the potential of aggregating opinions for multi-objective optimization problems. We show that agent teams are able to find a significantly larger percentage of optimal solutions than the current state of the art in building design. Hence, our application is able to lead to better designs, by: (i) providing the designer with a higher number of optimal solutions to choose from; (ii) eliminating suboptimal solutions falsely shown as "optimal" by individual agents; (iii) increasing the confidence of the designer that the system found the true pareto frontier.

When using multiple agents, we ran into the challenge of deciding how to form the team. It has been shown that diverse teams are able to outperform uniform teams composed by copies of the best agent in some scenarios (Marcolino, Jiang, and Tambe 2013; Marcolino et al. 2014). Hence, we also compare different teams, and we find that for some design problems, a diverse team outperforms a uniform team.

Related Work

This work is the result of a multi-disciplinary research, related to artificial intelligence and architectural design. We start by discussing AI works, and then we will present the state of the art of applying AI methods to building design.

In AI, this work is related to the study of team formation, social choice, ensemble systems, and genetic algorithms (GA). We begin by discussing team formation. Marcolino, Jiang, and Tambe (2013) and Marcolino et al. (2014) study teams of agents that vote together for solving complex problems. They show that for certain problems, a diverse team (composed by multiple different agents) is able to outperform a uniform team, composed by copies of the best agent.

Although the models presented are general, their experimental results are limited to the Computer Go domain. Besides, they consider only single-objective optimization problems.

Concerning social choice, many recent works 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) study 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 not yet "real world" ones. 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: architectural design. Moreover, social choice theory has not yet considered multi-objective optimization problems.

In machine learning, the aggregation of opinions has been widely studied. Generally, multiple weak learners vote together in classification problems (Polikar 2012). In our case, however, we are not dealing with a classification problem, but rather finding solutions to a complex multi-objective optimization problem by using evolutionary algorithms.

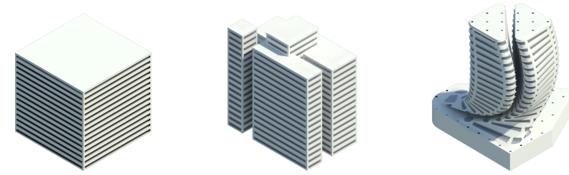
Distributed GA systems have also been studied (Knysh and Kureichik 2010). Our approach relates to the "island model", where populations evolve concurrently. Normally, however, the populations interact by transferring offsprings. To the best of our knowledge, aggregating GAs in the way that is explored in this work has not been tried before.

In the design studies, automated methods that can provide a high number of optimal alternatives are highly desirable, as it is hard for the human designers to manually find optimal solutions, and they need a large solution pool in order to be able to pick one that best fits their aesthetical/subjective evaluation (Flager, Gerber, and Kallman 2014; Lin and Gerber 2014; Welch, Moloney, and Moleta 2014; Briscoe 2014). The most common method for generating alternatives is to use genetic algorithms, as shown by a very recent and through survey of the literature (Zavala et al. 2014). However, there is not any work in design exploring the potential of agent teams to maximize the number of optimal solutions.

Design Domain

In the early stage of design, designers explore alternatives before proceeding to further design development. A broad range of possible solutions are intuitively and to a limited degree empirically analyzed. Typically energy performance assessments are made after this initial phase, where the analysis is performed on a very limited set of design alternatives (Radford and Gero 1980). Currently there is limited feedback between the domains of design and energy simulation available during the early stage. However, it has been acknowledged that such feedback has the highest potential impact on the building performance (Bogenstttter 2000; Lin and Gerber 2014).

In this work we use the Beagle system (Gerber and Lin 2013). Beagle is a multi-objective design optimization software framework that assists users in the early stage design of buildings. It incorporates an optimization methodology that combines parametric designs with multi-objective optimization through an integrated platform; enabling rapid iteration and trade-off analysis across different factors.



(a) Base (b) Office Park (c) Contemporary

Figure 1: Parametric designs used in our experiments.

First, the designer uses Autodesk Revit software to create a parametric design. This serves as a schematic design 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 design. The parameters can be integers or floating point numbers. These parameters, their type and their valid range are specified by the designer.

A series of designs have been studied in Gerber and Lin (2013) and a sub-set of these are used in this paper (Figure 1). We start with *base*, a simple building type with uniform program (i.e., tenant type). Then, in order to test the applicability of the method we progress to more real world like designs, with increased geometric as well as programmatic complexity (multiple tenant or functional occupancy types), leading to more challenging energy profiles for evaluation. Our second case study, *office park*, is a multi-tenant grouping of towers. The third, *contemporary*, is a double "twisted" tower that includes multiple occupancy types, relevant to contemporary architectural practices. These scenarios serve as proxies for real world design scenarios.

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 the Net Present Value, defined as follows.

SPCS defines how well a building conforms to the project requirements (by measuring how close the area dedicated to different activities is to a given specification). Let $\mathbf{L}$ be a list of activities (in our designs, **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 building floor area. It is calculated by the DOE-2.2 software, using the Autodesk Green Building Studio (GBS) web service (`https://`

gbs.autodesk.com/GBS/).

Finally, **NPV** is a commonly used financial evaluation. It 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. There will be a set of optimal (1^{st} ranked) solutions, and a designer must choose one among them according to his/her own subjective qualitative and quantitative evaluation. Note, however, that these are not necessarily the true optimal solutions of the multi-objective optimization problem, but merely the solutions that were not dominated by any other solution found by the GA.

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. More information can be found at Gerber and Lin (2013).

In the original Beagle, there is a deterministic algorithm for generating the initial population: the range of a parameter is uniformly divided among all offsprings. For example, if a parameter is an integer varying from 1 to 10 and the initial population size is 10, the first offspring will have value 1 for that parameter, the second value 2 and so on. In this work we modified the system, as the initial population would always be the same across different runs with the same initial population size. We randomized the initial population procedure: we keep the uniform division of the parameter range, but randomly decide which offspring will receive which value. We perform this modification in order to make the runs more randomized, avoiding that agents end up picking solutions sets that are too similar. The crossover and mutation also occurs probabilistically in the system.

Aggregating Opinions

We model each GA as an agent, that given a parametric design outputs its opinion ("optimal" solutions, according to its internal evaluation) about how the final design should be. We, then, aggregate the opinions of a team of agents. Since we are solving a multi-objective optimization problem, each agent has multiple optimal solutions. Therefore, we choose a limited number of them from each agent, and aggregate across all the combinations, generating a set of solutions for the team. Note that one combination includes only one solution from each agent, but different combinations will select different solutions from each agent.

Formally, let $\mathbf{\Phi}$ be a set of agents ϕ_i, and $\mathbf{\Omega}$ a set of parameters ω_j. Each ω_j has an associated set of possible values $\mathbf{A_j}$. Each agent ϕ_i is going to output a set $\mathbf{S^i}$ of solutions S_k^i, where a solution assigns a value a_j^i for each parameter ω_j. The solutions in $\mathbf{S^i}$ are optimal (1^{st} ranked, in a pareto ranking approach) according to the agent's internal evaluation, but they are not necessarily true optimal solutions.

Hence, given a fixed S_k^i for each agent ϕ_i, we can aggregate all a_j^i, creating a new solution of the team S_k^T. In this paper we study three different aggregation methods: *mean:* for each ω_j, calculates the mean across all a_j^i; *median:* for each ω_j, calculates the median across all a_j^i; *vote:* for each ω_j, assigns the value a_j that appears the most often among all a_j^i (ties are broken randomly). We consider values that are the same up to 3 decimal places as equal.

In order to obtain the solutions S_k^i, we consider a subset $\mathbf{S'^i} \subset \mathbf{S^i}$ containing only n solutions for each agent ϕ_i. We can then select one solution S_k^i from each $\mathbf{S'^i}$, and aggregate those to create one solution S_k^T. By repeating this process through all the possible combinations of one solution from each $\mathbf{S'^i}$, we create a set of team solutions $\mathbf{S^T}$.

The system allows each agent to be initialized with different options, which affects the agent behavior. Hence, we can select different options for each ϕ_i, or set the same options for all ϕ_i. The first approach will generate a *diverse* team, and the second a *uniform* team. Note that the agents of the *uniform* team will not necessarily output the same set of solutions, since the search is stochastic and they are initialized with different random seeds.

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

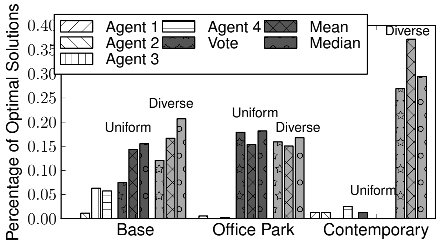
Table 1: GA parameters for the diverse team. Initial Population and Maximum Iteration were kept as constants: 10 and 5, respectively. PZ = Population Size, SZ = Selection Size, CR = Crossover Ratio, MR = Mutation Ratio.

Experiments

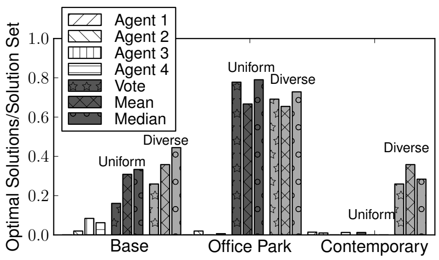
We run experiments across the different parametric designs shown in Figure 1. These are designs with increasing complexity. More details about the designs and the meanings of each parameter are available in Gerber and Lin (2013). We create 4 different agents, using the options in Table 1.

In this work we are dealing with real (and consequently complex) design problems. Hence, for these parametric designs the true set of optimal solutions are not known in advance. Therefore, we approach the problem in a comparative fashion: when evaluating different systems, we consider the union of the set of solutions of all of them. That is, if each system x has a set of solutions $\mathbf{H}_x$, we consider the set $\mathcal{H} = \bigcup_x \mathbf{H}_x$. We, then, compare all solutions in $\mathcal{H}$, and consider as optimal the best solutions in $\mathcal{H}$, forming the set of optimal solutions $\mathcal{O}$. We use here the concept of pareto dominance. That is, the best solutions in $\mathcal{H}$ are the ones that *dominates* all other solutions (i.e., they are better in all 3 factors). As we know which system generated each solution $o \in \mathcal{O}$, we are able to estimate the number of unique optimal solutions of each system under evaluation.

We first compare all solutions of all agents (i.e., construct the $\mathcal{H}$ set as the union of the solutions of all agents), in order to estimate which one has the largest set of optimal solutions. We then run again that agent multiple times, creating the *uniform* team. Next, we aggregate the solutions of the

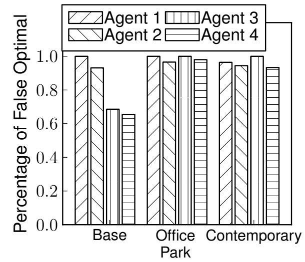


(a) Percentage in relation to all solutions found by all systems.

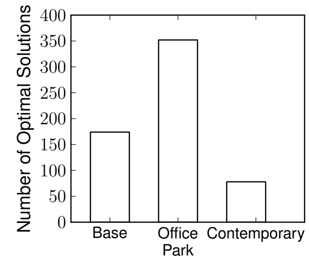


(b) Percentage in relation to the number of solutions evaluated by each system.

Figure 2: Percentage of unique optimal solutions found by each system



(a) False optimal solutions that are eliminated.



(b) Number of unique optimal solutions, by all agents and teams.

Figure 3: Additional analysis.

diverse and the *uniform* team. We use $n = 3$, that is, we use three solutions from each agent.

We evaluate together all the solutions of the agents and the teams (i.e., we construct the $\mathcal{H}$ set with the solutions of all systems), in order to estimate the number of unique optimal solutions of each system. Since the true optimal solutions set is unknown, we first plot in Figure 2 (a) the percentage of unique solutions found by each system in relation to the total number of unique optimal solutions found in $\mathcal{H}$ ($|\mathcal{O}|$).

In all parametric designs the *diverse* team is able to find a larger percentage of optimal solutions than the individual agents. In general, the agents find less than 5% of the solutions (with a few exceptions that go slightly above 5%), while the *diverse* team is always close to or above 15%. Note that, in total (considering all aggregation methods), for *base* the agents are able to find only 13% of the optimal solutions, while *uniform* finds 37% and *diverse* 49%. For *office park*, the agents find merely 0.8%, while *uniform* find 51%, and *diverse* 47%. Finally, for *contemporary*, the agents find only 5%, while *diverse* finds 93% and *uniform* 1% of the optimal solutions. As we can see, *diverse* is able to find a larger percentage of solutions than *uniform* in the *contemporary* and in the *base* design. We are currently analyzing why the *uniform* team has such a low performance for *contemporary*.

In Figure 2 (b), we show the percentage of optimal solutions, in relation to the size of the set of evaluated solutions of each system. That is, let $\mathbf{O}_x$ be the set of optimal solutions of system x, in $\mathcal{O}$. We show in the figure the value $\frac{|\mathbf{O}_x|}{|\mathbf{H}_x|}$. Concerning *vote*, for example, the teams are able to find a new optimal solution around 20% of the time for *base*, around 73% of the time for *office park* and around 25% of the time for *contemporary* (considering the *diverse* team). Meanwhile, for the individual agents the number is close to 0% (except for agent 3 and 4 in *base*, that have around 7%).

One of the advantages of our approach is that it allows the designer to eliminate solutions that are falsely reported to be optimal by the individual agents. Hence, in Figure 3 (a) we show the percentage of solutions that were reported to be optimal by each individual agent, but were later discovered to be suboptimal by evaluating the set with all solutions $\mathcal{H}$. We can see that a large amount of solutions are eliminated (in some cases close to 100% of the reported solutions), helping the designer to avoid making a poor decision, and increasing her confidence that the set of optimal solutions found represent well the "true" pareto frontier.

Moreover, we test for duplicated solutions across different aggregation methods, different teams and different agents. We find that there are no duplicates in all parametric designs. Hence, we obtain a high coverage of the pareto frontier. We show the total number of unique optimal solutions in Figure 3 (b).

Finally, in order to better study the solutions proposed by the agents and teams, we show all the optimal solutions in the factors space in the online appendix (at `http://teamcore.usc.edu/people/sorianom/wcs15-ap.pdf`), where we show that the solutions give a good coverage of the pareto frontier.

Conclusion

We showed the potential of aggregating the opinions of a team for architectural building design. That is a very important application in the current scenario, since the designs are optimized according to energy performance (among other factors). Our approach provides a significantly larger percentage of optimal solutions than the current state of the art, enabling the designer to make a better decision according to her subjective evaluation. Moreover, we eliminate falsely reported optimal solutions by single agents, increasing the designer confidence in finding the true pareto frontier.

Acknowledgments: This research is supported by MURI grant W911NF-11-1-0332, and the National Science Foundation under grant 1231001.

References

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

Briscoe, D. 2014. Parametric planting: Green wall system research + design using bim. In *Proceedings of the Association for Computer-Aided Design in Architecture 2014 International Conference*, ACADIA.

Caragiannis, I.; Procaccia, A. D.; and Shah, N. 2013. When do noisy votes reveal the truth? In *Proceedings of the 14th ACM Conference on Economics and Computation*, EC, 143–160.

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* 90(8):936–959.

Knysh, D. S., and Kureichik, V. M. 2010. Parallel genetic algorithms: a survey and problem state of the art. *Journal of Computer and Systems Sciences International* 49(4):579–589.

Lin, S.-H. E., and Gerber, D. J. 2014. Evolutionary energy performance feedback for design: Multidisciplinary design optimization and performance boundaries for design decision support. *Energy and Buildings* 84:426–441.

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, 1142–1148.

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 the 28th Conference on Artificial Intelligence*, AAAI, 1485–1491.

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, 279–285.

Polikar, R. 2012. Ensemble learning. In Zhang, C., and Ma, Y., eds., *Ensemble Machine Learning: Methods and Applications*. Springer.

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

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 25th Conference on Neural Information Processing Systems*, NIPS, 126–134.

Welch, C.; Moloney, J.; and Moleta, T. 2014. Selective interference: Emergent complexity informed by programmatic, social and performative criteria. In *Proceedings of the Association for Computer-Aided Design in Architecture 2014 International Conference*, ACADIA.

Zavala, G. R.; Nebro, A. J.; Luna, F.; and Coello, C. A. C. 2014. A survey of multi-objective metaheuristics applied to structural optimization. *Structural and Multidisciplinary Optimization* 49:537–558.