Spring 1-1-2017

Numerical Service Life Model of Chloride Induced Corrosion in Recycled Aggregate Concrete and Design Optimization of Sustainable and Durable Concrete Mixtures

Nathan Daniel Stambaugh
University of Colorado at Boulder, nast1697@colorado.edu

NUMERICAL SERVICE-LIFE MODEL OF CHLORIDE-INDUCED CORROSION IN RECYCLED AGGREGATE CONCRETE AND DESIGN OPTIMIZATION OF SUSTAINABLE AND DURABLE CONCRETE MIXTURES

by:

NATHAN DANIEL STAMBAUGH

B.S. University of Colorado Boulder, 2016

A thesis submitted to the

Faculty of the Graduate School of the

University of Colorado in partial fulfillment

of the requirement for the degree of

Master of Science

Architectural Engineering

2017

This thesis entitled:
Numerical Service Life Model of Chloride Induced Corrosion in Recycled Aggregate Concrete and Design Optimization of Sustainable and Durable Concrete Mixtures

written by Nathan Daniel Stambaugh

has been approved
for the Department of Civil, Environmental, and Architectural Engineering

_____
Prof. Wil V. Srubar III

_____
Prof. Joseph Kasprzyk

_____
Prof. Yunping Xi

Date_____

The final copy of this thesis has been examined by the signatories, and we find that both the content and the form meet acceptable presentation standards of scholarly work in the above mentioned discipline

Stambaugh, Nathan Daniel (M.S., Architectural Engineering)
Numerical Service Life Model of Chloride Induced Corrosion in Recycled Aggregate Concrete and Design Optimization of Sustainable and Durable Concrete Mixtures

Thesis directed by Assistant Professor Wil V. Srubar III

Concrete mixture design is a multi-objective optimization problem that often involves the weighing of trade-offs between fresh-state properties, such as workability, flowability, and set time with hardened-state properties, such as strength, freeze thaw durability, and chloride durability. Substantial time and effort goes into designing, testing, and perfecting single mixture designs for optimal performance in some, but not all, of these properties. Additionally, considerations for mixture cost and environmental impacts, such as embodied energy, and embodied carbon are often included only as a secondary objective. The complex relationships between concrete mixture proportions and resulting constituent properties are investigated herein through two studies that explore mixture design tradeoffs.

The first study presents the development, validation, and implementation of a 1D numerical service-life prediction model for reinforced recycled aggregate concrete exposed to internal and external sources of chlorides. The model accounts for the inclusion of supplementary cementitious materials (SCMs), namely (a) fly ash, (b) slag, (c) silica fume, and (d) metakaolin, and recycled aggregates (i) with and (ii) without initial chloride contamination from previous in-service exposure. The model is used to predict time to corrosion-induced cracking for reinforced recycled aggregate concrete in five case-study applications, namely structures in a marine splash zone (Zone I), a marine spray zone (Zone II), within 800 km of coastline (Zone III), within 1.5 km of coastline (Zone IV), and parking structures at

locations greater than 1.5 km from the coastline (Zone V) in Los Angeles, California and Anchorage, Alaska. The effects of recycled aggregate size, replacement ratio, degree of aggregate pre-contamination with chloride from previous in-service exposure, water-to-cement (w/c) ratio, and SCMs on time-to-cracking of reinforced recycled aggregate concrete are elucidated herein. The potential for SCMs to improve the service life of recycled aggregate concrete is investigated by estimating theoretical additions required to meet a target service life of 50 years.

Results indicate that, in addition to geographic location, temperature, and severity of chloride exposure, w/c ratio and aggregate replacement ratio exhibit the greatest impact on time to chloride-induced cracking in reinforced recycled aggregate concrete. Furthermore, initial aggregate chloride contamination and aggregate size impart minimal effects on expected service life. Finally, the results illustrate that the use of either fly ash or slag is most viable in achieving a 50-year service life for recycled aggregate concretes in chloride-laden environments.

Broadening the work of the first study, the second study presents the development and implementation of a multi-objective concrete mixture design tool to evaluate tradeoffs of different mixture proportions on the physical, mechanical, and environmental performance of concrete. The model utilizes a multi-objective evolutionary algorithm (MOEA) that uses a search-based methodology to find a set of Pareto-optimal mixture designs. Mathematical relationships informing the MOEA consider the effect of cement content, water, supplementary cementitious materials (SCMs), namely (i) fly ash, (ii) silica fume, (iii) slag and (iv) metakaolin, sand, coarse aggregate, recycled aggregates, and air on fresh- and hardened-state properties, cost, and environmental impacts. Six objectives are used to determine optimality of mixtures: strength, workability, chloride induced corrosion resistance, embodied energy, embodied carbon, and cost. Objective properties are modeled

using a suite of empirical and numerical methods that consider multiple decisions in their formulation. As demonstrated herein, the model can produce a suite of optimal mixtures for three case study applications, namely (1) a cubic meter of concrete, (2) a tilt-up concrete wall, and (3) a concrete column. Up to 10 total design scenarios are investigated for each case study to illustrate the capabilities of the MOEA modeling methodology.

Results indicate that a MOEA design approach can elucidate tradeoffs related to mixture proportions, transportation cost, constituent cost, and performance. Additionally, the results illustrate that multiple design constraints, including simultaneous consideration of durability and strength criteria, can be implemented in the model and produce a set of optimal, viable solutions. Finally, the need for additional refinement of the relationships that inform the MOEA optimization as well as additional objectives and decisions to approach a more holistic design procedure are discussed.

ACKNOWLEDGMENTS

I would first like to thank my two research advisors, Dr. Wil V. Srubar III and Dr. Joseph R. Kasprzyk, for their combined vision and expertise that led to the successful completion of this work. Additionally, I would be remiss if I did not thank Dr. Srubar individually for his enduring support and mentorship; without which I would have achieved far less. I am beyond grateful for the life-altering skills and experiences that he has passed on to me. Dr. Kasprzyk also deserves individual praise for his willingness and ability to aid me in completing this work on short notice. Finally, I would like to thank my third committee member, Dr. Yunping Xi, for his time, consideration, and expertise in reviewing my work within such a short time.

Special thanks are due to Mikaela DeRousseau for her friendship throughout our mutual struggle to get up to speed on the NSF project to which we both belong. I would also like to thank those members of the SIMLab group, past and present, who, directly through their research or indirectly through friendship, have contributed to this work. They are Adriana Souto, Todd Bergman, Juan Pablo Gevaudan, Elizabeth Delesky, Anastasia Aday, Kristen Hess, Sarah Hong, and Jorge Osio-Norgaard.

Lastly, I would like to thank the wonderful CEAE staff and faculty that have kept my time at CU on a steady upward trajectory. I am especially appreciative of the work done by Erin Jerick and Araceli Warren who were critical in both my undergraduate and graduate success. Without all of these people and the contributions of my family and friends this last year as a graduate student and previous four years as an undergrad would not have been filled with such happiness and growth.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1 INTRODUCTION

Worldwide, concrete is the most common building material and the second most consumed material on Earth after water [1]. Subsequently, its production, use, and disposal have global environmental consequences. The production of cement alone is responsible for 5-8% of anthropogenic carbon dioxide emissions, which exacerbates effects related to global warming and climate change [2]. In addition, debris generated by the demolition of concrete structures is a large contributor to industrial waste streams. While using recycled concrete in low-performance applications is a common practice, the use of crushed recycled concrete as aggregate in new structural concrete remains uncommon, primarily due to a lack of confidence in mechanical properties [3], [4] and appropriate modeling tools to predict long-term performance. Often, the inclusion of recycled aggregates in a concrete mixture is justified as contributing to the sustainability of the mixture because it is considered a recycled component. Current mixture design methodologies, however, have no framework for determining if recycled aggregates (which reduce the physical and mechanical properties of concrete) are truly a sustainable edition to a mixture.

In order to reduce the impact of concrete carbon emissions, which is attributable primarily due to the production of cement, many concrete producers replace portions of cement content with supplementary cementitious constituents (SCMs). SCMs such as fly ash and slag, byproducts form the burning of coal and manufacturing of steel, respectively, have been shown to reduce the environmental impacts of ordinary portland cement (OPC) concrete [5]–[7]. Additionally, SCMs have been shown to improve mechanical and durability properties of concrete with

increased replacement of cement [8]–[10]. Although utilized in many mixtures, the choice and quantity of SCMs is often determined solely based on impact on strength and constituent availability. Furthermore, the negative impacts of SCMs at high replacement percentages, namely on concrete workability, are not always considered in initial mixture proportioning and are often countered with high additions of chemical admixtures, i.e. superplasticizers, that can have high embodied impacts.

While inclusion of recycled aggregates and SCMs have both positive and potentially negative consequences, there does not yet exist a holistic decision-making framework for their inclusion and optimal proportioning in a concrete mixture. Tradition methods, like the American Concrete Institute's (ACI) absolute volume method (AVM) addresses both of these constituents as a secondary consideration to the target design strength of a concrete mixture. Without incorporating the effect of recycled aggregates and SCMs on concrete performance, proper use of each constituent is limited to the experience of the concrete producer or structural engineer. A better understanding of the impacts of recycled aggregates, as well as a new mixture design methodology that holistically considers the intricate relationships between the type and proportions of constituent materials on concrete performance, would, for the first time in decades, advance concrete mixture design methodology.

This thesis contains two independent studies on the modeling and design of concrete mixtures. The first, **Chapter 2**, is on the development, validation, and implementation of a 1D numerical service life prediction model for recycled aggregate concrete. The model utilizes the Crank-Nicolson finite difference method as a numerical solution to Fick's Second Law of Diffusion to model chloride ion

ingress through 1D media. Time, temperature, and inclusion of SCMs and recycled aggregates (up to 100%) on the diffusivity of concrete are considered.

The second, **Chapter 3**, presents the development and implementation of a novel multi-objective optimization tool for concrete mixture design. Optimal concrete mixtures are determined by a multi-objective evolutionary algorithm (MOEA) named Borg. Objectives in the model include cost, embodied energy, embodied carbon, strength, workability, and durability. The model utilizes empirical and statistical relationships to predict desired objectives while considers not only the inclusion of recycled aggregates and SCMs, but also the in situ sequesterable carbon. Optimal mixtures are considered to be those that minimize all objectives with the exception of strength and workability, which are maximized.

The specific goals of the first study were to:

- Develop a numerical 1D chloride diffusion model that accounts for increased diffusivity due to the incorporation of recycled aggregates;
- Evaluate the importance of climate (i.e., geographic location) and exposure on the chloride resistance of recycled aggregate concrete;
- Elucidate the impact of w/c ratio, initial recycled aggregate chloride contamination (from previous exposure), recycled aggregate replacement ratio, and aggregate size on the service-life of recycled aggregate concrete;
- Analyze trade-offs between the use of SCMs, namely fly ash, silica fume, metakaolin, and slag, on recycled aggregate concrete service life;
- Evaluate the capability of fly ash, silica fume, metakaolin and slag to achieve a desired (target) service life of 50 years for a recycled aggregate concrete exposed to a range of exposure conditions in two geographic locations.

The specific goals of the second study were to:

- Develop a model capable of optimizing concrete mixture designs with regard to multiple performance objectives;
- Calibrate the model with industry data and ensure reasonable, practical solutions;
- Evaluate the model's ability to produce solution sets for a variety of case study applications (e.g., cubic meter, tilt-up wall, column);
- Elucidate Illustrate, for the first time, decision trade-offs in the design of sustainable and resilient concrete mixtures.

CHAPTER 2 NUMERICAL SERVICE LIFE MODELING OF CHLORIDE INDUCED CORROSION IN RECYCLED AGGREGATE CONCRETE

2.1. INTRODUCTION

Chloride-induced corrosion is one of the most common durability issues encountered by reinforced concrete structures. Laboratory and modeling studies have consistently shown that the chloride permeability of recycled aggregate concrete increases with aggregate replacement ratio due to increases in average pore size and total concrete porosity[11]–[15]. Other experimental studies have shown that chloride resistance can be further compromised by initial contamination of recycled aggregates from previous in-service exposure [16], [17].

To improve confidence in the long-term durability of both normal and recycled aggregate concrete, engineers require suitable modeling tools to estimate the service-life performance of concrete structures in chloride-laden environments. Despite being widely used, service-life prediction tools, such as Life-365 [18] and STADIUM® [19], do not yet account for the use of recycled aggregates. However, several researchers have proposed models to predict chloride transport in recycled aggregate concrete, including the authors, who previously proposed the first steady-state model for chloride diffusion in both contaminated- and non-contaminated recycled aggregate concrete. Xiao, et al., [20]used the finite element method to conduct a parametric study that elucidated the effects of aggregate replacement ratio, shape, boundary conditions, and attached mortar on the effective diffusivity of recycled aggregate concrete. Ying, et al., [21] proposed a new model that described the effects of recycled aggregate distribution on chloride diffusion. Srubar [17]

proposed a steady-state 1D solution to Fick's Second Law of Diffusion that accounted for pre-contamination of recycled aggregates. The model was based on dopant-diffusion principles in which a substrate is doped with concentrations of another constituent that diffuse throughout the bulk over time. Despite this modeling advance for contaminated recycled aggregate concrete, incapability of accounting for non-steady-state boundary conditions and time-dependent changes in the chloride diffusion coefficient is a limitation of the approach.

2.1.1. Scope of work

To address the limitations of the previously proposed steady-state chloride diffusion model for reinforced recycled aggregate concrete, this chapter presents the formulation, validation, and implementation of a 1D numerical finite difference solution to Fick's Second Law of Diffusion that is used with a simplified cracking model to predict time to corrosion-induced cracking in contaminated and non-contaminated recycled aggregate concrete. The model, which is based on the finite difference solution employed in Life-365 [18], accounts for non-steady-state chloride boundary conditions, recycled aggregate size, placement, replacement ratio, and initial degree of contamination, and effects of water-to-cement (w/c) ratio, time, temperature, and supplementary cementitious constituent (SCM) additions on the chloride diffusion coefficient. The numerical model is first validated with Life-365 using normal aggregate concrete and is subsequently enhanced with the most current parametric relationships and implemented using a stochastic approach to estimate time to corrosion-induced cracking of recycled aggregate concrete in five case-study applications, namely structures in a marine splash zone (Zone I), a marine spray zone (Zone II), within 800 km of coastline (Zone III), within 1.5 km of coastline (Zone IV) and parking structures (Zone V) in Los Angeles, California and Anchorage, Alaska. In addition to elucidating the effects of key modeling

parameters, the amount and type of SCM required to meet a target service life of 50 years in each case-study application is investigated herein.

2.2. MODEL DEVELOPMENT

The time to corrosion-induced cracking was estimated using a two-part damage model first proposed by Tuutti [22]. The model considers the total service life, t_s, of reinforced recycled aggregate concrete the sum of two successive time periods, namely, time-to-corrosion-initiation, t_i, which is governed by the diffusion of chlorides throughout the concrete media, and the time-to-corrosion-cracking, t_c. In the following section the mathematics of the time-to-corrosion initiation model is described based on how it is utilized in the model. Additionally, the effects of water/cement ratio (w/c), SCM addition, time and temperature on the diffusion coefficient is covered. Boundary conditions that are considered to ramp up with time to a max concentration are defined and are the last section in **Section 2.2** pertaining to the time-to-corrosion initiation model. Next, the model and coefficients used to predict time to cracking are presented completing the definition of the time-to-corrosion-cracking model used in this study. The final two sections discuss all of the modeling parameters used within the model and the procedural process by which all the modules discussed in **Section 2.2** are implemented.

2.2.1. Time-to-corrosion-initiation

Time-to-corrosion-initiation is a function of the transport properties of the concrete, geometry, the boundary conditions that exist for a given environment and application, and the required concentration of chlorides to initiate the corrosion of the reinforcing steel (i.e., chloride threshold). Corrosion initiation is defined as the time that it takes for chlorides from the surrounding environment to penetrate the concrete cover and accumulate to a sufficient concentration at the reinforcement

surface to initiate corrosion. Chloride concentrations above the chloride threshold locally reduce the pH near the reinforcement, which results in depassivation of the protective oxide layer and subsequent corrosion of the steel reinforcement.

Chloride transport can take place due to a number of mechanisms including (a) diffusion under the influence of a concentration gradient, (b) absorption due to capillary action, (c) migration in an electrical field, and (d) pressure-induced flow and wick action when water absorption and water vapor diffusion are combined [23]. Ionic diffusion of chloride is the primary mechanism of chloride transport and is considered the sole mechanism for the models discussed in this study. It has been shown that the relationship between chloride concentration, diffusion coefficient, and time in the random molecular motions of chloride ions in concrete can be described using Fick's Second Law of Diffusion [24], a governing second-order partial differential equation that is used to characterize the diffusion process:

$$\frac{\partial C}{\partial t} = -D\frac{\partial^2 C}{\partial x^2} \qquad \text{Eq. 2-1}$$

where C is the chloride concentration (kg/m^3), D is the apparent diffusion coefficient (m^2/s), x is the depth from the exposed concrete surface (m), and t is time (s).

A finite difference method is employed to numerically solve the diffusion equation and estimate time-to-corrosion initiation. The approach is based on the well-known Life-365 service-life prediction software [18], which utilizes the Crank-Nicolson finite difference method to numerically solve the diffusion equation for normal aggregate concrete. The Crank-Nicolson approach is implemented here to account for limitations of the simple error function solution previously reported by Srubar [17], namely its inabilities to account for non-steady state boundary conditions and time-dependent changes in chloride diffusion coefficients.

According to the method, a 1D representation of the concrete cover depth is divided into a finite number of slices, s, and nodes, n, where $n = s + 1$ as illustrated in **Figure 2-1**. The chloride concentration at an arbitrary node, i, at a timestep of t+1 is calculated by the advection-dispersion equation:

$$-ru_{i+1}^{t+1} + (1+2r)u_i^{t+1} - ru_{i-1}^{t+1} = ru_{i+1}^{t} + (1-2r)u_i^{t} + ru_{i-1}^{t} \qquad \text{Eq. 2-2}$$

where the dimensionless Courant-Friedrichs-Lewy (CFL) number:

$$r = D_t\left(\frac{dt}{2(dx)^2}\right) \qquad \text{Eq. 2-3}$$

and where $D_{t,T}$ is the diffusion coefficient (m^2/s) at time, t, and temperature, T, d_t is the timestep (s), dx is the length of each slice (m), $u^{t}{}_{i}$ is the chloride concentration (kg/m^3) at the node at time, t. The left side of Eq. 2-2 represents unknown concentrations at a future timestep, where the right side of the equation represents known values at the current timestep. The surface chloride concentration and the time-dependent constituent properties (i.e., diffusion coefficient) of the concrete are calculated at the beginning (and held constant) during each timestep.

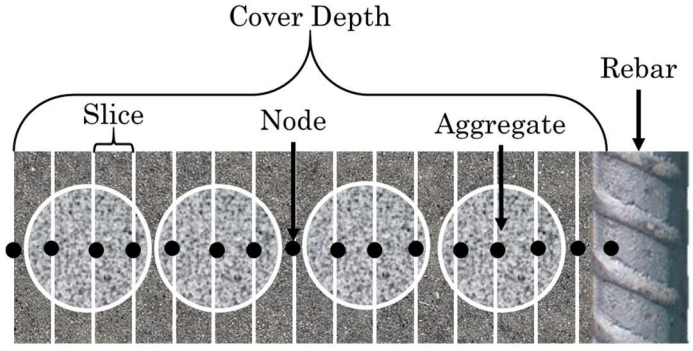

Figure 2-1. Graphical Representation of the 1D numerical Approach

To numerically solve the diffusion equation, Eq. 2-2 can be rearranged into matrix form:

$$AU^{t+1} = BU^{t} \qquad \text{Eq. 2-4}$$

where A and B are dimensionless matrices corresponding to the dimensionless CFL number:

$$A = \left\{a_i^{t+1}\right\} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ -r & 1+2r & -r & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & -r & 1+2r & -r \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$B = \left\{b_i^{t+1}\right\} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ r & 1-2r & r & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & r & 1-2r & r \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \qquad \text{Eq. 2-5}$$

Given that the chloride diffusivity of recycled aggregates (which primarily consist of old mortar) is not similar to the diffusivities of impermeable normal aggregates or new cement paste, the diffusion coefficients of recycled aggregates were modeled by setting the CFL number, r, in A and B that corresponded with the nodes representing the location of the recycled aggregates to a constant that more accurately represented the higher chloride diffusivity of recycled aggregates (D_a = 12.5 ± 2.0 x 10^{-12} m^2/s). See **Section 2.3** for a summary of modeling parameters.

The U^t matrix defines the chloride concentration of each node throughout the cover depth at the present timestep:

$$U^t = \left\{ u_i^t \right\} = \begin{bmatrix} u_1^t \\ \cdots \\ u_n^t \\ \cdots \\ u_n^t \end{bmatrix} \qquad \text{Eq. 2-6}$$

The node $u^t{}_1$ corresponds with the chloride concentration at the surface of the concrete, which was set at the start of each simulation and, subsequently, at the start of each iterative timestep according to the boundary condition models described in **Section 2.1.5**. Pre-contamination of recycled aggregates was modeled by setting initial chloride concentration of each node in U^t that corresponded with the location of a randomly placed recycled aggregate equal to the degree of initial aggregate contamination (see **Section 2.3**). This procedure was performed only for the initial timestep, $t = 0$, allowing the initial chloride contamination to diffuse through the bulk.

Using matrix inversion, Eq. 2-4 can be rearranged to solve for U^{t+1}, a matrix of chloride concentrations for each individual slice at the next timestep:

$$U^{t+1} = \left\{ u_i^{t+1} \right\} = \begin{bmatrix} u_1^{t+1} \\ \cdots \\ u_n^{t+1} \\ \cdots \\ u_n^{t+1} \end{bmatrix} \qquad \text{Eq. 2-7}$$

In the iterative numerical simulation, the U^{t+1} matrix thusly becomes the U^t matrix for the following timestep, and only the first term in the matrix need be updated to reflect the change in the surface boundary condition prior to stepping through the next iteration. See **Section 2.1.5** for boundary condition modeling.

Since the Crank-Nicolson finite difference method accounts for both space and time, it is possible to included non-steady-state transport properties (i.e.,

chloride diffusion coefficients) and boundary conditions. Therefore, the effects of (1) w/c ratio, (2) SCMs, (3) time, and (4) temperature on the bulk apparent diffusion coefficient, (5) time-dependent boundary conditions, and the potential for (6) initial contamination of recycled aggregates were included in the modeling methodology. Mathematical incorporation of each of these effects is discussed in the following sections.

2.2.2. Effect of w/c ratio

The w/c ratio is well known to impact chloride diffusion coefficients. Per the model proposed in Riding et al., [10], the effect of w/c ratio on the chloride diffusion coefficient was described by:

$$D_{28} = 2.17 \times 10^{-12} e^{(w/c)/0.279} \quad \text{Eq. 2-8}$$

where D_{28} is the 28-day diffusion coefficient (m^2/s), and w/c is in decimal form.

2.2.3. Effect of SCMs

Previous research has shown that incorporating SCMs modifies many constituent properties of concrete, including chloride resistance. As proposed in [10], [25], three SCMs, namely silica fume (SF), fly ash (FA), and metakaolin (MK), were considered to affect D_{28} according to:

$$D_{SF} = D_{28} \cdot (0.206 + 0.794e^{(-SF/2.51)}) \quad \text{Eq. 2-9}$$

$$D_{UFFA} = D_{28} \cdot (0.170 + 0.829e^{(-UFFA/6.07)}) \quad \text{Eq. 2-10}$$

$$D_{MK} = D_{28} \cdot (0.191 + 0.809e^{(-MK/6.12)}) \quad \text{Eq. 2-11}$$

where D_{SF}, D_{FA}, and D_{MK} are the modified 28-day diffusion coefficients due to the addition of *SF*, *FA*, and *MK*, respectively. *SF*, *FA,* and *MK* are the percent replacement (in whole-number percent) of ordinary portland cement.

Additionally, two SCMs, namely FA and SG, alter the decay rate, m, of the diffusion coefficient when accounting for time (see **Section 0**) according to [19]:

$$m = 0.26 + 0.4\left(\frac{FA}{50} + \frac{SG}{70}\right) \quad \text{Eq. 2-12}$$

where FA and SG are again the whole-number percent replacement of ordinary portland cement by FA and SG, respectively.

SCMs also impact fresh-state workability, set time, and early strength gain of concrete when used in excessive amounts [26], [27]. Thus, SCM additions in this model were limited to the replacement values listed in **Table 2-1** that do not require corrective viscosity-modifying or set-accelerating admixtures.

Table 2-1. Maximum replacement ratios of SCMs used in model implementation.

Allowable Replacement of SCMs		
SCM	Max Replacement	Source
Silica fume	10%	[26], [28]
Fly ash	25%	[28], [29]
Metakaolin	20%	[27], [30]
Slag	50%	[31], [32]

2.2.4. Effect of time

The chloride diffusion coefficient is a time-dependent parameter that is well known to decrease with time. This reduction is due, in part, to continued cement hydration and densification of the concrete beyond the first 28 days, among other mechanisms (i.e., carbonation). At each timestep, the bulk diffusivity was recalculated according to the following relationship:

$$D_t = D_{28}\left(\frac{t_{28}}{t}\right)^m + D_{ult}\left(1-\left(\frac{t_{28}}{t}\right)^m\right) \quad \text{Eq. 2-13}$$

where D_t is the time-dependent diffusion coefficient, D_{28} is either the 28-day diffusion coefficient calculated by Eq. 2-8 (without SCM addition) or Eq. 2-9-11(with SCM addition), and the reference time, t_{28}, is typically taken as 28 days. As discussed, previous work [10] suggests that the chloride diffusion coefficient eventually plateaus to a final value. The second term of Eq. 2-13 accounts for this plateau effect *via* D_{ult}, which is the 100-year ultimate diffusion coefficient calculated by the first term in Eq. 2-13.

2.2.5. Effect of temperature

Temperature is an important consideration for a non-steady-state diffusion model as it can change the rate of concrete densification as well as the rate of chloride ion diffusion. To account for temperature-dependence of the chloride diffusion coefficient at each timestep, a simple Arrhenius relationship was implemented in this model [18]:

$$D_{tT}(t,T) = D_t(t)\cdot\exp\left[\frac{U}{R}\left(\frac{1}{T_{ref}}-\frac{1}{T}\right)\right] \quad \text{Eq. 2-14}$$

where $D_{t,T}$ is the time- and temperature-dependent diffusion coefficient (m^2/s) at every node at each timestep throughout the cover depth, since temperature equilibrium was assumed to be reached immediately in comparison to chloride equilibrium, U_a is the activation energy (J/mol), R is the universal gas constant (8.3144 J/mol/K), T is the average monthly temperature for the location of interest (K), and T_r is the reference temperature equal to 20°C (294.15 K).

2.2.6. Chloride boundary condition modeling

Time-dependent chloride boundary conditions were modeled identically to the bilinear ramp-up and plateau models implemented in Life-365. The ramp-up phase simulates chloride build-up due to cycles of wetting and drying at the concrete surface [18]. The slope of the ramp-up and the subsequent plateau of the chloride concentration at the concrete surface depend on both geographic location and exposure classification. **Table 2-2** summarizes the boundary condition modeling parameters used in this study for each of the five exposure classifications. Given their similar coastal geographies, the boundary condition modeling parameters were identical for all exposure classifications in both geographic locations investigated in this study, namely Los Angeles, California (CA) and Anchorage, Alaska (AK).

Table 2-2. Parameters for boundary condition modeling from Life-365.

Zone	Exposure Classification	Chloride Concentration	
		Ramp-up (years)	Maximum (wt.%)
I	Marine Splash Zone	1	0.8
II	Marine Spray Zone	10	1
III	Within 800 m of Coastline	15	0.6
IV	Within 1.5 km of Coastline	30	0.6
V	Parking Structure	200	0.8

2.2.7. Time to cracking

Time-to-cracking is defined as the time from corrosion initiation to stress-induced cracking of the concrete cover based on the time-dependent formation of oxidation products. Time-to-cracking was estimated using a model proposed by Liu

and Weyers [33], which has been used in many studies to predict the cracking in reinforced concrete structures in chloride-laden environments [34]–[36]. According to the model, t_c, can be predicted according to:

$$t_c = \frac{W_{crit}^2}{2k_p} \quad \text{Eq. 2-15}$$

where W_{crit} is the amount of corrosion products required to cause cracking and k_p is the rate of rust production. W_{crit} can be computed according to the following:

$$W_{crit} = \rho_r \left(\pi \left[\frac{Cf'_c}{E_{eff}} \left(\frac{a^2 + b^2}{a^2 - b^2} + \upsilon \right) + t_p \right] d_b + \frac{W_{st}}{\rho_{st}} \right) \quad \text{Eq. 2-16}$$

where x_c is the concrete cover depth, ρ_r and ρ_s are the density of rust and steel, respectively, f'_t is the tensile strength of concrete (MPa), υ is Poisson's ratio, t_p is the thickness of the porous zone surrounding the steel reinforcement, d_b is the diameter of the reinforcement, and E_{eff} (MPa) is the effective elastic modulus of concrete modified by a creep coefficient, φ.

$$E_{eff} = \frac{E}{1 + \varphi} \quad \text{Eq. 2-17}$$

The variables a and b are the inner and outer diameter of an idealized concrete cylinder of expansion (mm), respectively:

$$a = \frac{d_b + 2t_p}{2} \quad \text{Eq. 2-18}$$

$$b = C + a \quad \text{Eq. 2-19}$$

The amount of corroded steel, W_{st}, is equal to $W_{st} = \alpha W_{crit}$, where α is the ratio of the molecular weight of steel and the molecular weight of rust products. The

values for α depend on the type of corrosion products and typically vary between 0.523-0.622 [33].

$$k_p = \alpha^{-1} d_b i_{corrr} \quad \text{Eq. 2-20}$$

Finally, the rate of rust production is described: where i_{corr} is the annual mean corrosion rate (A/m^2).

2.2.8. Modeling parameters

Modeling parameters used in all numerical simulations are presented in **Table 2-3**. Means and standard deviations of all physical-based parameters were identified from lab or field measurements reported in literature. Deterministic parameters are shown as single values. Statistically distributed parameters are presented with a mean ± standard deviation.

Table 2-3. Fixed modeling parameters.

Service-life modeling parameter	Value	Units	Refs.
Time-to-corrosion-initiation			
Cover Thickness, x_c	70 ± 5	mm	[37], [38]
Recycled aggregate diffusion coefficient, D_a	12.5 ± 2.0	kg/m^3	[17], [20]
Chloride threshold, c_t[a]	0.7 ± 0.05	kg/m^3	[39]–[41]
Time to corrosion cracking			
Tensile strength, f'_t	3.75 ± 0.5	MPa	[11]
Modulus of elasticity, E	30 ± 3.0	GPa	[11]
Phi (creep coefficient), ϕ	2	-	[33]
Poisson's ratio, v	0.18	-	[33], [42]
Density of rust, ρ_r	3600	kg/m^3	[33], [43]
Density of steel, ρ_s	7850	kg/m^3	[33], [43]
Thickness of porous region, t_p	12.5 ± 0.5	µm	[33], [44]
Corrosion rate, i_{corr}	2.5 ± 0.5	$\mu A/cm^3$	[33], [43]
Alpha, α[b]	0.523-0.622	-	[33]
Mild steel rebar diameter, d_b	9.5	mm	

[a] Lognormally distributed,

[b] Uniformly distributed.

Case study structures in two geographic locations were considered in this analysis. For each zone and location, four parameters (see **Table 2-4**) were varied to investigate the effects each had on the expected service life of recycled aggregate

concrete. Initially, a default value for each of the four parameters was set as a baseline that were chosen to be representative of intermediate conditions or values that would be common in practice. Systematic analyses were conducted by varying a single parameter per simulation while holding the rest to default values.

Table 2-4. Variable modeling parameters. Default values were held constant in simulations where others were varied to elucidate effects on expected service-life of recycled aggregate concrete.

Parameters	Default Values	Investigated Values	Units	Refs.
Water-to-Cement (w/c) Ratio	0.45	0.30 0.35 0.40 0.45	-	[45]
Aggregate pre-contamination	2.0	1.00 1.50 2.00 2.50	kg/m^3	[46]
Aggregate diameter	9.5	9.50 12.7 19.0 25.4	mm	[47]
Aggregate replacement ratio	0.5	0.30 0.50 0.70 1.00	-	[45]

2.2.9. Numerical simulation procedure

In the numerical simulation, first, geographic location and exposure zone were defined, which determined average monthly temperature profiles and parameters for the chloride boundary condition modeling (**Table 2-2**), respectively. This study explicitly investigated two locations, namely Los Angeles, CA and Anchorage, AK, whose average monthly temperature profiles are shown in **Figure 2-2**.

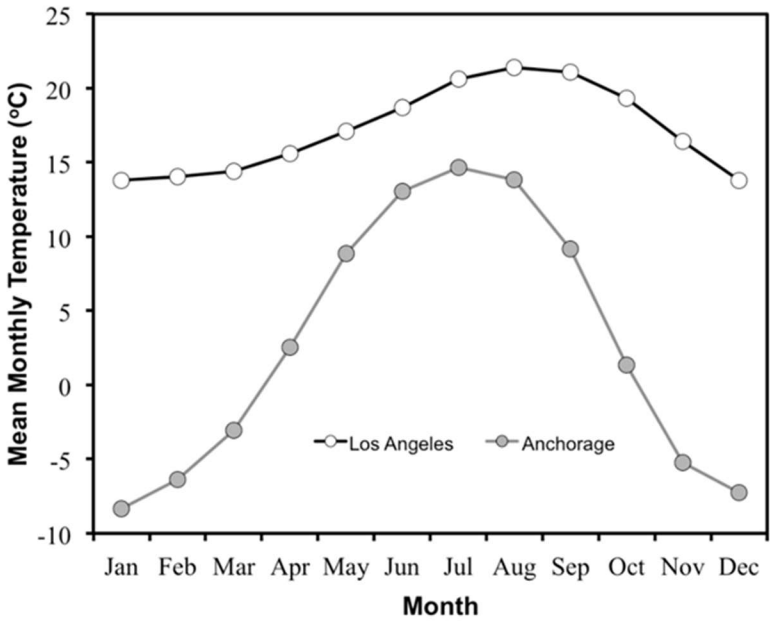

Figure 2-2. Average monthly temperature profiles for Los Angeles, CA and Anchorage, AK.

Monte Carlo analysis was used to incorporate uncertainty to account for in-field variation in concrete cover depth, chloride threshold, and constituent properties *via* a simple bootstrap method similarly employed in [37] in which modeling parameters that were statistically distributed were first sampled from assumed distributions for each simulation. A normal distribution was assumed for all parameters unless otherwise noted. Therefore, once geographic location and zone were defined, the fixed time-to-corrosion-initiation modeling parameters (**Table 2-2**) were sampled from their respective statistical distributions, along with deterministic variables that were explicitly under investigation and directly defined by the user (**Table 2-4**). Aggregate placement was then randomly generated throughout the cover depth by (1) dividing the cover depth by maximum aggregate size into aggregate zones (to the nearest lower-bound integer) and (2) randomly placing an aggregate within each of those zones assuming a uniform distribution of placement while ensuring no aggregate overlap. A statistical test was used to

designate aggregates as normal or recycled aggregates, according to the replacement ratio explicitly defined by the user (**Table 2-3**). As discussed in **Section 2.2.1**, the CFL number for recycled aggregates was fixed for the entire simulation for those nodes that corresponded to the placement of recycled aggregates. Else, the chloride diffusion coefficient was used for each node throughout the cover depth, including those nodes corresponding to normal aggregates, as similarly employed by Life-365.

The model then calculated the w/c- and SCM-modified baseline (D_{28}) diffusion coefficient (Eq. 2-8-11). At each subsequent timestep, a new time- and temperature-dependent chloride diffusion coefficient was determined according to Eq. 2-13 and Eq. 2-14, respectively, and the new chloride distribution throughout the cover depth was calculated for the next timestep according to matrix inversion (Eq. 2-7). Timesteps were taken to be one-month increments, and the cover was discretized into 300 equal slices (s=300, n=301). After each timestep, if the concentration at the cover depth was greater than or equal to the randomly generated chloride threshold from **Table 2-3**, corrosion initiation was assumed to have occurred for that simulation, and the model would store this time as time-to-corrosion-initiation, t_i.

Once time-to-corrosion-initiation was calculated, time-to-cracking, t_c was computed directly using Eq. 2-15 with randomly sampled inputs from parameters listed in **Table 2-3**. Each computed service life, $t_s = t_i + t_c$, was then stored and the stochastic process was repeated. By an analysis of variance, a total of 20,000 Monte Carlo simulations were required to yield statistically significant predictions for expected service life of recycled aggregate concrete.

2.3. RESULTS AND DISCUSSION

Results in this chapter are broken into 8 subsections. The first section is devoted to the validation of the normal aggregate concrete model and the impact of the more updated empirical relationships, particularly those discussed in **2.2.3**. climatic effects, namely, temperature are presented next. The proceeding four sections present results of the stochastic model used to identify impacts of the selected model parameters discussed in **2.2.8**. Finally, the impact of fixed SCM replacement is considered follow by an investigation into the SCM replacement needed for recycled aggregate mix to attain a 50-year service life.

2.3.1. Model validation

The proposed numerical model was first validated *via* comparison with Life-365 using normal aggregate concrete with the parameters listed in **Table 2-5**. Given that Life-365 has no stochastic capability, comparisons were made using deterministic values. For validation purposes, results were obtained for both a numerical model that employed an identical input for a constant decay factor (m = 0.2) that is used in the current version of Life-365 [18] (Numerical Model A) and the numerical model proposed herein (Numerical Model B) that included the most recent relationships for m (Eq. 2-12), the diffusion coefficient plateau (Eq. 2-13), and the effects of w/c and SCMs on D_{28}, as presented in **Section 2.2**.

The results of the model comparisons are shown in **Table 2-6**. The results indicated that the numerical model was formulated identically to the numerical model used by Life-365. Minor differences (≤0.5%) between the results obtained by Numerical Model A and Life-365 were determined to be a result of Life-365 accounting for leap years. The differences between Numerical Model B and Life-365 are attributable to the updated mathematical relationships for m and chloride diffusion coefficients, which are, incidentally, less conservative than the equations

used in the current version of Life-365. Thus, the results obtained using Numerical Model B were used for the remainder of the analyses presented in this work.

Table 2-5. Input parameters for model validation and comparison with Life-365.

Thickness (mm)	Cover Depth (mm)	Location	D_{28} (m^2/s)	Chloride Threshold (%)
500	70 mm	Los Angeles, CA	1.1×10^{-11}	0.05

Table 2-6. Service-life prediction model validation and comparison with Life-365.

Zone	Life-365 (Years)	Numerical Model A (Years)	Mode A Difference	Numerical Model B (Years)
I	4.6	4.6	0.0%	5.8
II	8.6	8.6	0.0%	10.4
III	12.9	12.9	0.0%	15.7
IV	18.2	18.3	0.5%	22.1
V	45.1	45.3	0.4%	53

2.3.2. Impact of climate

To establish a normal aggregate concrete baseline, the effect of climate, namely average monthly temperature, on time-to-cracking of normal aggregate concrete (w/c=0.45) in all exposure classifications is elucidated by the results presented in **Figure 2-3**. Averaged data correspond to a 50% likelihood of corrosion-induced cracking as determined by the stochastic approach implemented by the numerical simulation. For example, in Anchorage, AK in Zone IV, there is a 50% likelihood that normal aggregate concrete will exhibit corrosion-induced cracking in

30.2 years, while, in Los Angeles, CA there is an identical likelihood that cracking will occur in 19.9 years for normal aggregate concrete placed in the same zone.

As anticipated, both location and chloride exposure conditions affect the expected service life of reinforced concrete. The service life of identical normal aggregate concretes exposed to the same boundary conditions is longer in colder climates (AK) than in warmer climates (CA) due to expected high temperature-dependent increases in the chloride diffusion coefficient. Similarly, as expected, more aggressive chloride exposure conditions in Zones I and II result in a reduced anticipated service life of normal aggregate concrete in comparison to lower chloride exposure conditions of Zones III, IV, and V in both geographic locations.

The results also illustrate that temperature-related increases in expected service life are more pronounced as chloride exposure conditions worsen. For instance, concrete placed in Zone I in Anchorage, AK exhibits a 65% increase in expected service life compared to the same concrete placed in Zone I in Los Angeles, CA. This temperature-related benefit decreases to 44% in Zone 5 exposure conditions, suggesting a more significant role of average temperature in extending the service life of concrete placed in high-exposure conditions.

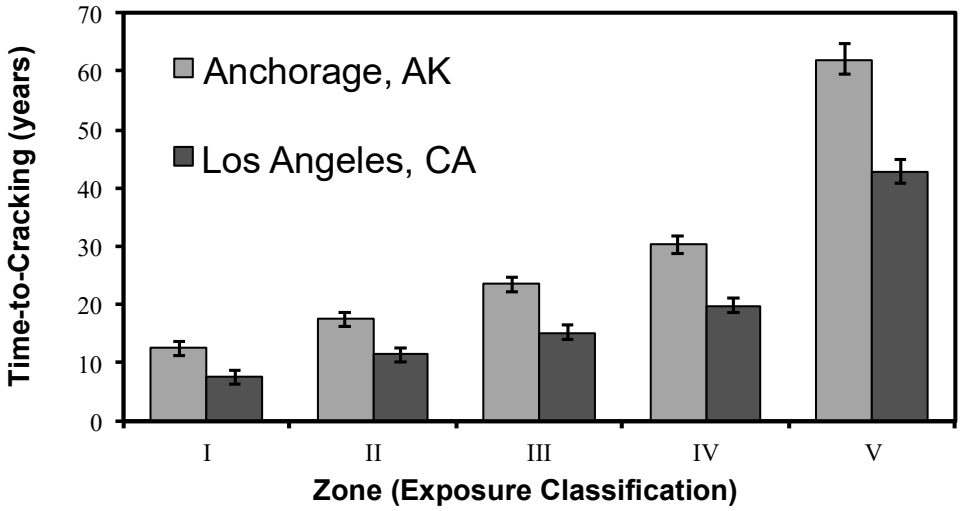

Figure 2-3. Baseline analysis. Temperature effects on expected service life of normal aggregate concrete in five exposure classifications in Anchorage, AK and Los Angeles, CA.

2.3.3. Impact of initial aggregate contamination

The effect of initial aggregate contamination on the service-life of recycled aggregate concrete in chloride-laden environments is shown in **Figure 2-4**. The results were obtained from an analysis that used the default input parameters for recycled aggregate concrete listed in **Table 2-4**, namely a $w/c = 0.45$, aggregate replacement ratio of 0.5, and an aggregate diameter of 9.5 mm, and the initial aggregate contamination level was varied in the model by each of the incremental values in **Table 2-4**. Averaged data correspond to a 50% likelihood of corrosion-induced cracking as determined by the stochastic approach implemented by the numerical simulation.

As in previous work [17], the results indicate that, while all concretes exhibit reductions in service life from the normal aggregate baselines shown in Figure 2-4, increases in initial aggregate contamination from 1.0-2.5 kg/m^3 result in little

variation in expected service life of recycled aggregate concrete. A 50% aggregate replacement and initial contamination decreased the service life of the recycled aggregate concretes by a maximum of 41% (in Los Angeles, CA) and a minimum of 23% (in Anchorage, AK) compared to the baseline case (**Figure 2-3**) for the recycled aggregate concretes and ranges of initial aggregate contamination that investigated in this study. Within each location and exposure condition, the difference between the lowest and highest contamination values decreased the expected service life by a maximum of 3.6% in Los Angeles, CA and 11.5% in Anchorage, AK, respectively. These results, however, were not statistically significant.

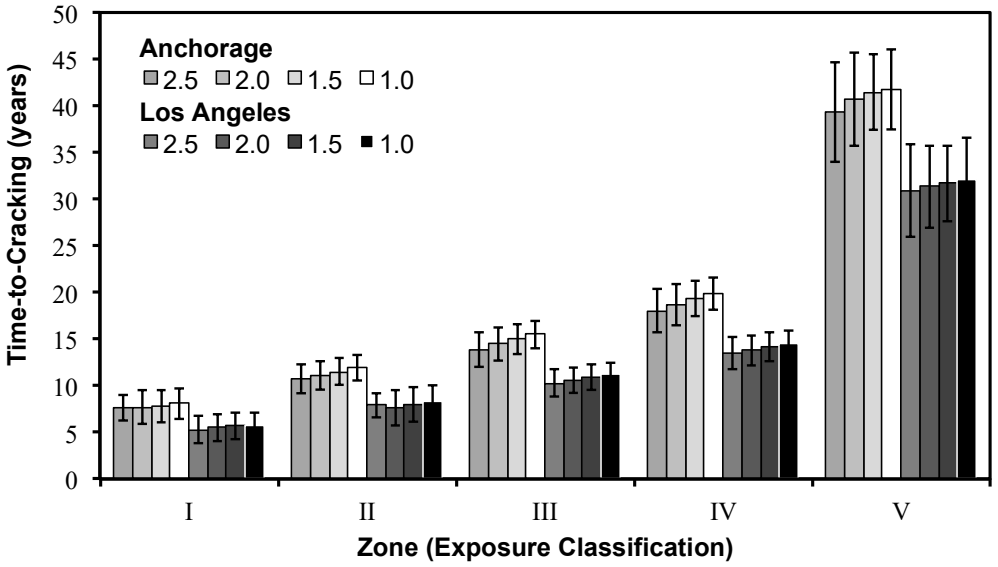

Figure 2-4. Effect of initial aggregate contamination (kg/m^3) on expected service life of recycled aggregate concrete in five exposure classifications in Anchorage, AK and Los Angeles, CA.

While these results are specific to a recycled aggregate concrete with a 50% aggregate replacement ratio, w/c ratio of 0.45, and 9.5 mm aggregate diameter, previous research has comprehensively shown that the effect of initial

contamination is minor in comparison to location and exposure conditions. Srubar [17] found that the effect of aggregate pre-contamination on expected time to cracking was less exaggerated in the presence of more aggressive external chlorides. These results also ascertain that external boundary conditions have a greater effect on in-service durability than do residual aggregate chloride contamination. Such findings suggest that certain levels of aggregate chloride contamination may be permissible in the design of recycled aggregate concrete structures in chloride-laden environments.

2.3.4. Impact of recycled aggregate size

The effect of aggregate size on the anticipated service-life of recycled aggregate concrete is shown in **Figure 2-5**. The results were obtained from an analysis that used the default input parameters for recycled aggregate concrete listed in **Table 2-4**, including an aggregate replacement ratio of 50% and an initial aggregate contamination of 2.0 kg/m^3, which was assumed to ensure that approximately the same number and type of contaminated aggregate would be in each simulation. Aggregate diameter was varied in the model by each of the incremental values listed in **Table 2-4**. Averaged data correspond to a 50% likelihood of corrosion-induced cracking as determined by the stochastic approach implemented by the numerical simulation.

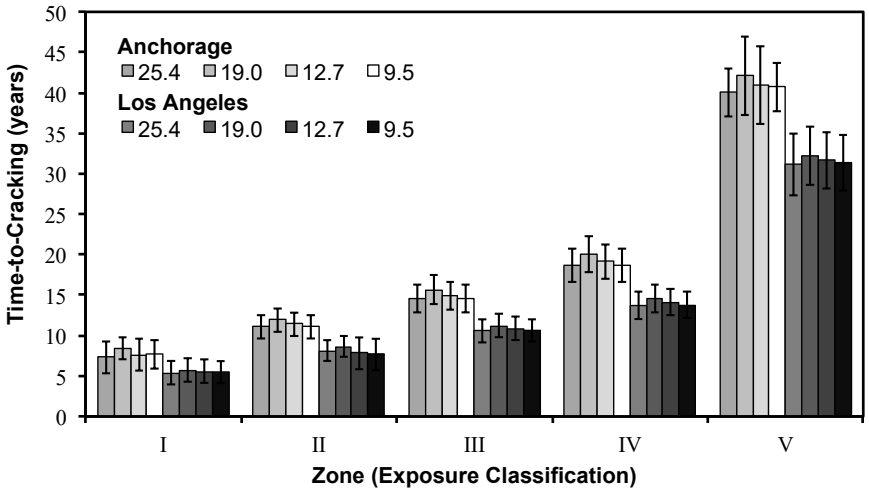

Figure 2-5. Effect of aggregate diameter (mm) on expected service life of recycled aggregate concrete in five exposure classifications in Anchorage, AK and Los Angeles, CA.

Similar to the effects of initial aggregate contamination, the results show that, despite initial reductions in expected service life compared to normal aggregate concrete, changes in aggregate diameter exhibit a negligible effect on the service life of contaminated recycled aggregate concrete. In contrast to the normal aggregate baseline case in **Figure 2-3**, the expected service life of recycled aggregate concretes decreased by a maximum of 42% and a minimum of 25%. However, this reduction was uniform across all exposure classifications and ranges of aggregate diameter investigated herein. The slight increase in service life observed for the 19 mm aggregates is due to the partitioning of the cover depth in the numerical model and a maximum placement of three instead of four aggregates in series. These findings further suggest that aggregate size does not impart a significant effect on the durability of recycled aggregate concrete structures, especially those in non-extreme chloride exposure conditions.

2.3.5. Impact of recycled aggregate replacement ratio

The effect of aggregate replacement ratio on the anticipated service-life of recycled aggregate concrete is shown in **Figure 2-6**. The results were obtained from an analysis that used the default input parameters for recycled aggregate concrete listed in **Table 2-4**, including an initial aggregate contamination level of 2.0 kg/m^3. Aggregate replacement ratios were varied in the model from 30-100% by the incremental values listed in **Table 2-4**. A 0% replacement was not investigated, since such an analysis already corresponded to the normal aggregate concrete baseline case presented in **Figure 2-3**. Averaged data correspond to a 50% likelihood of corrosion-induced cracking as determined by the stochastic approach implemented by the numerical simulation.

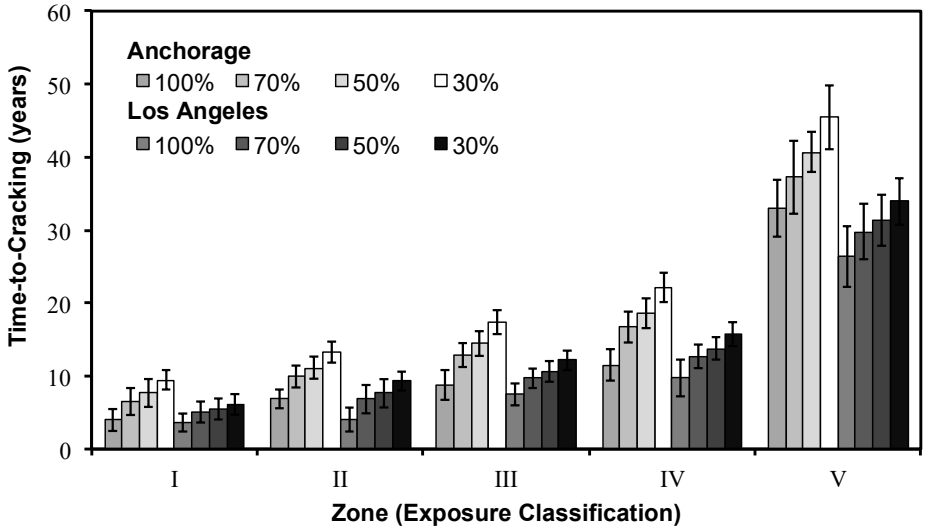

Figure 2-6. Effect of recycled aggregate replacement ratio on expected service life of recycled aggregate concrete in five exposure classifications in Anchorage, AK and Los Angeles, CA.

As anticipated, the results show that an increase in aggregate replacement ratio corresponds with a decrease in anticipated service life. Compared to the baseline case (**Figure 2-3**), increasing the replacement ratio from 30% to 100% leads to a minimum and maximum reduction in expected service life of 18% (in Los Angeles, CA) and 68% (in Anchorage, AK), respectively. Within each exposure classification, the reductions are most pronounced in high-exposure conditions. For example, expected service life decreased 57% with increases in recycled aggregate replacement ratios from 30% to 100% in tidal zones (Zone I) in Anchorage, AK. In a low-exposure application (Zone V), expected service life of recycled aggregate concrete with identical recycled aggregate replacement ratios decreased only 22% in Los Angeles, CA. Previous research on the effect of recycled aggregate replacement ratio on mechanical and durability properties of concrete has indicated that at all levels of replacement are admissible, depending on the desired application, design life, and anticipated service life of recycled aggregate concrete [48], [49]. The results from this analysis further substantiate this conclusion, given the range (6.2-45.5 years) of expected service life for recycled aggregate concrete. For some structural applications with design lives greater than 30 years, medium levels of pre-contamination (2.0 kg/m^3) and high aggregate replacement ratios may be acceptable in some applications where mild levels of surface chloride exposure are anticipated. However, further analyses would need to be conducted to reveal absolute limits of aggregate replacement ratios and initial contamination given specific design scenarios in certain geographic locations.

2.3.6. Impact of w/c ratio

Figure 2-7 shows the effects of w/c ratio on the expected service life of recycled aggregate concrete. Results were obtained from an analysis that also used the default input parameters for recycled aggregate concrete listed in **Table 2-4**,

including an aggregate replacement ratio of 50%, a constant aggregate diameter (9.5 mm), and an initial aggregate chloride contamination of 2.0 kg/m^3. For this analysis, the w/c ratio was varied in the model by the incremental values listed in **Table 2-4** to elucidate the effects of w/c ratio on expected service life. Averaged data correspond to a 50% likelihood of corrosion-induced cracking as determined by the stochastic approach implemented by the numerical simulation.

As expected, an increase in w/c ratio results in a decrease in expected service life of recycled aggregate concrete in both locations across all exposure classifications investigated in this study. Increases in w/c ratio are well known to affect total porosity, chloride permeability, and, thus, expected service life of reinforced concrete in chloride-laden environments [11]–[13], [46], [50]. Increasing the w/c from 0.30 to 0.45 results in a minimum and maximum reduction in expected service life of 9% (in Los Angeles, CA) and 38% (in Anchorage, AK), respectively, in comparison to the normal aggregate concrete baseline case (see **Figure 2-3**). This result can be attributed to even higher porosities and lower chloride resistances expected of recycled aggregate concrete in comparison to normal aggregate concrete.

Similar to increases in aggregate replacement ratio, the reductions in expected service life due to increases in w/c ratio are most pronounced in high-exposure environments. For instance, **Figure 2-7** illustrates that the expected service life of recycled aggregate concrete in Anchorage, AK decreases 29% and 22% in Zone I and Zone V, respectively, when w/c is varied from 0.30-0.45. Similar, yet less severe, reductions in Zone I (20%) and Zone V (18%) are observed for Los Angeles, CA. Together with the previously discussed results, these data suggest that, in addition to location, anticipated chloride exposure, and aggregate replacement ratio, w/c ratio remains an important durability parameter in

estimating the service life of recycled aggregate concrete in chloride-laden environments.

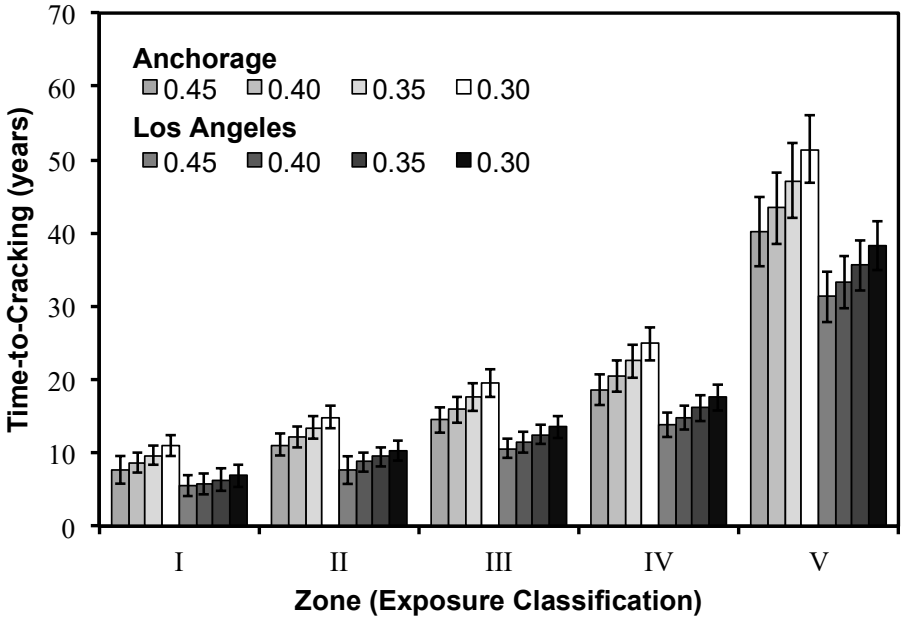

Figure 2-7. Effect of w/c ratio expected service life of recycled aggregate concrete in five exposure classifications in Anchorage, AK and Los Angeles, CA.

2.3.7. Impact of SCM additions

The effect of 5% SCM addition on the expected service life of recycled aggregate concrete is shown in **Figure 2-8**.Results were obtained from an analysis that used the input parameters in the default case (DC) scenario for recycled aggregate concrete listed in **Table 2-4**, including a 50% recycled aggregate replacement ratio, initial aggregate contamination, a $w/c = 0.45$, and an aggregate diameter of 9.5 mm, and 0% SCMs. A fixed replacement of 5% for each SCM, namely fly ash (FA), slag (SG), silica fume (SF), and metakaolin (MK), was assumed for the comparative simulations and modeled according to the relationships presented in **Section 2.2.2**. The 5% cement replacement was chosen for illustrative

purposes so that no SCM would be near its maximum advisable replacement limit set forth in **Table 2-1**. Averaged data correspond to a 50% likelihood of corrosion-induced cracking as determined by the stochastic approach implemented by the numerical simulation.

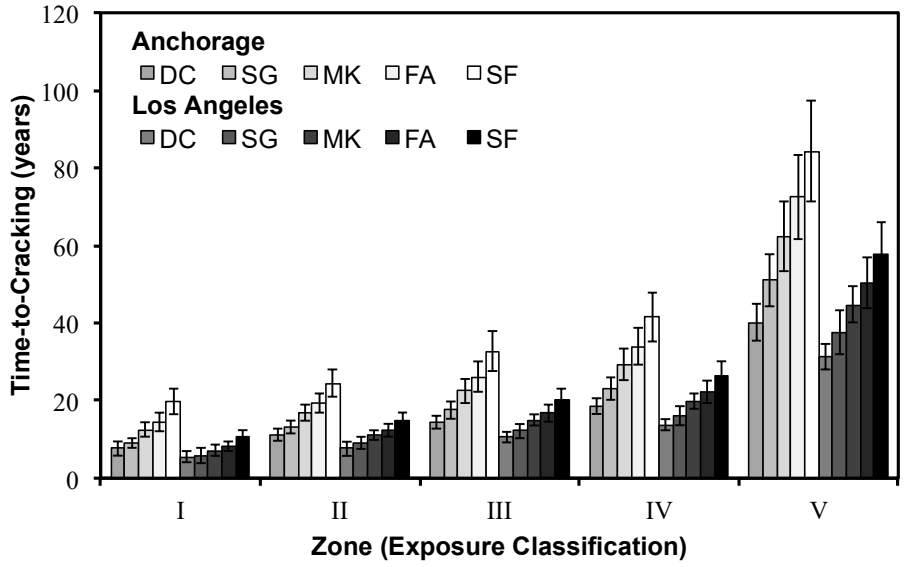

Figure 2-8. Effect of 5% SCM replacement (by mass) on expected service life of recycled aggregate concrete five exposure classifications in comparison to the default case (DC).

The results in **Figure 2-8**reiterate that the inclusion of recycled aggregates in the default case scenario (DC) reduce chloride resistance and, subsequently, expected service life in contrast to normal aggregate concrete (**Figure 2-3**). However, the results also indicate that additions of SCMs improve the performance of recycled aggregate concrete in chloride-laden environments compared to the recycled aggregate DC baseline. Extensive experimental research has previously shown that the inclusion of supplementary cementitious constituents (SCMs) improves concrete mechanical properties and resistance to chloride ingress [9], [10].

These modeling results also substantiate that SCMs improve the expected service life of recycled aggregate concrete.

According to **Figure 2-8**, even with a moderate replacement (5%) of ordinary cement, SCMs can overcome the initial reduction in service life that was expected of recycled aggregate concrete and, in some cases, supersede the service life of non-SCM normal aggregate concrete. For example, the expected service life of recycled aggregate concrete with 5% FA and 5% SF in Anchorage, AK exceed that of normal aggregate concrete (**Figure 2-3**) by 11-17% and 36-58%, respectively in Zones I-V. Similar, yet less substantial, increases in service life are observed for 5% replacements of FA and SF in all Zones in Los Angeles, CA.

The results also illustrate that SF and FA are more effective than metakaolin and slag in extending the anticipated service life of recycled aggregate concrete in small additions. This result can be attributed to high silica contents and, hence, increased pozzolanic reactivity, of silica fume and fly ash in comparison to slag and metakaolin. The effect of the pozzolanic reaction, namely the reaction between soluble silica and calcium hydroxide that produces dense calcium-silicate-hydrate, on the bulk transport properties is taken into account in the modeling by the modifying the effective diffusion coefficients calculated according to Eq. 2-9-11.

In addition, the data suggest that benefits of 5% SCM additions on the expected service life of recycled aggregate concrete are both location- and application-dependent. With 157% and 96% increases in expected service life over the recycled aggregate concrete DC scenario, the use of 5% SF resulted in the most benefit in Zone I and Zone II applications in Anchorage, AK and Los Angeles, CA, respectively. The use of 5% FA, however, resulted in a maximum of 87% and 61% increases in expected service life over the recycled aggregate concrete DC scenario in Zone I and Zone IV applications in Anchorage, AK and Los Angeles, CA,

respectively. Such results demonstrate the location- and application-dependencies of the effects that different SCMs have on the expected service life of recycled aggregate concrete.

2.3.8. SCM addition for 50-year service life

Previous experimental research has shown that the addition of SCMs in recycled aggregate concrete could result in concrete strength and durability performance equivalent to that of normal aggregate concrete [51], [52]. However, the exact replacements needed for equal performance is uncertain due to the high variability in recycled aggregate quality and SCM chemistries investigated by these studies.

To better estimate required SCM replacements for equivalent service life performance, the data shown in **Table 2-7** are the result of a hypothetical analysis that used the default input parameters for recycled aggregate concrete (DC) listed in **Table 2-4** to investigate the type and quantity of SCMs required to reach a target 50-year design life in the five case-study applications in both Los Angeles, CA and Anchorage, AK. The model was implemented deterministically (i.e., no statistical sampling) utilizing a fixed aggregate position based on the specified aggregate replacement ratio (50%), aggregate size (9.5 mm), degree of initial aggregate contamination (2.0 kg/m^3), and w/c ratio (0.45).

The data in **Table 2-7** show the difference in performance among the four SCMs investigated herein. All values reported as N/A indicate that required SCM replacements exceeded 100% and were, thus, unattainable. The other values are hypothetical results that were not bound by the practical limitations listed in **Table 2-1**. Lower values required for identical applications in Anchorage, AK in comparison to Los Angeles, CA are again indicative of the effect of geographic location and elevated temperatures on exacerbating chloride diffusivity. Thus,

higher SCM replacement percentages are required to meet the target design life for the same application in warmer climates.

Table 2-7. Hypothetical type and quantity of SCM (%) replacements needed to reach target 50-year service life in five exposure classifications in Anchorage, AK and Los Angeles, CA.

	SF		FA		SG		MK	
Zone	AK	LA	AK	LA	AK	LA	AK	LA
I	N/A	N/A	16	25	70	94	N/A	N/A
II	N/A	N/A	14	23	61	75	N/A	N/A
III	34	N/A	11	18	52	72	N/A	N/A
IV	12	N/A	9	14	41	61	N/A	N/A
V	1	3	3	7	6	21	5	26

The results suggest that FA and SG, not SF or MK, are most beneficial in increasing the expected service life of the recycled aggregate concrete DC scenario to 50 years, especially in high-exposure environments. According to the results, a replacement of 16% and 25% FA would reach a target design life of 50 years in Zone I exposure classifications in Anchorage, AK and Los Angeles, CA, respectively. Both replacement percentages of FA are well within the practical limits set forth in **Table 2-1**. Similarly, the percent replacements for SG are within reason, especially for Zone III-V exposure classifications in Anchorage, AK. Neither MK nor SF were as effective at improving the service life of the recycled aggregate concrete investigated in this particular analysis using the default parameters listed in **Table 2-4**. However, practical percent replacements of MK or SF could be achievable in

extending the service life of other recycled aggregate concretes in high-exposure environments with different design parameters.

2.4. CONCLUSIONS

This work presented the development, validation, and implementation of a 1D numerical service-life model for reinforced recycled aggregate concrete, which overcomes limitations of current service-life prediction models, including inabilities to account for fluctuating boundary conditions, time- and temperature-dependent chloride diffusion coefficients, and pre-contaminated recycled aggregates. Using a probabilistic approach, the model was first validated with the service-life prediction software Life-365 and subsequently enhanced and employed to predict the time to corrosion-induced cracking of recycled aggregate concrete using aggregates (i) with and (ii) without initial chloride contamination from previous in-service exposure. Additionally, the impacts of location were investigated by temperature effects and variation of exposure conditions, namely structures located in a marine splash zone (Zone I), a marine spray zone (Zone II), within 800 km of coastline (Zone III), within 1.5 km of coastline (Zone IV) and parking structures (Zone V) in Los Angeles, California and Anchorage, Alaska. The effects of (a) degree of initial aggregate contamination, (b) aggregate size, (c) aggregate replacement ratio, and (d) water-to-cement (w/c) ratio were investigated, as were the effects of (e) including supplementary cementitious constituents (SCMs), namely fly ash, slag, silica fume and metakaolin, were included in the model formulation.

The results from the subsequent analyses illustrate that, similar to normal aggregate concrete, exposure conditions, namely temperature and chloride exposure, exhibit the greatest impact on expected service life of recycled aggregate concrete. Results also show that increases in both aggregate replacement ratio and

w/c ratio decrease expected service life. However, aggregate size and initial aggregate contamination levels were not found to affect expected service life in the applications and exposure conditions investigated herein. In regard to SCMs it was found that silica fume exhibited the most noticeable effect at low-percent replacements, while practical additions of fly ash and slag were the most viable in achieving a desired service life of 50 years in all five exposure cases for the recycled aggregate concrete investigated in this study.

The results indicate that moderate- to high-replacement ratios and some level of initial aggregate contamination may be permissible in the design of recycled aggregate concrete structures in chloride-laden environments and that the size of contaminated aggregates should not necessarily take priority as a durability design consideration. Analyzing the use of SCMs elucidated their importance, especially fly ash, if recycled aggregate concrete is to be used in applications with decades-long exposure to chlorides. Finally, the results indicate that recycled aggregate concrete appears to be best suited for cooler climates where service life can be prolonged simply through the retardation effect that lower temperatures have on chloride ingress.

This chapter broadly indicates that the quality and quantity of constituents used in a concrete mixture greatly impact the durability of the resulting constituent. This chapter indicates that the key to ensuring a desired service life for concrete does not involve optimizing any one parameter. Instead, it eludes to the idea that, although some parameters have greater impacts than others, there is no ideal solution for all applications. Indeed, as **Table 2-7** indicates there may be multiple acceptable, useful and potentially optimal solutions to a concrete with recycled aggregates. This potential for recycled aggregate use as well as other trade-offs are explored in **Chapter 3**.

CHAPTER 3 DESIGN OPTIMIZATION OF SUSTAINABLE AND DURABLE CONCRETE MIXTURES

Results of **Chapter 2** indicated that concrete mixtures are highly susceptible to reduced durability performance with the inclusion of recycled aggregates. Additionally, location was shown to play a significant role in mixture design to ensure adequate durability. However, durability (i.e., anticipated service life) was the only factor in justifying the benefits of a non-recycled aggregate mixture over those with recycled aggregates. Considerations for cost, environmental impacts, strength, regional availability, and workability (among other important considerations) were not included. Without a holistic approach to evaluate the impact of concrete constituents on performance, a mixture design that optimally balances multiple competing design considerations is not achievable.

Over the past century, useful mathematical relationships have been developed by previous researchers that link concrete mixture proportions to fresh- and hardened-state properties, such as strength, workability, and durability. Combining these relationships to design a single mixture enables a more holistic—and optimal—design process than traditional mixture design methods (i.e., Absolute Volume Method).

The design of optimal concrete mixtures inherently involves the resolution of conflicting performance goals, i.e. strength and cost, which leads to no single optimal design, but a suite of possible, equally viable, alternatives. Thus, the utility of an optimal mixture design tool lies in its ability to narrow the set of potential mixtures to only the most optimal choices, which can then be prioritized by the

designer according to individual preferences. Such a design process could not only save on time, but also increase the efficiency and diversity of concrete mixtures utilized by industry.

3.1. INTRODUCTION

While concrete mixture design optimization has been utilized for single objective problems, the implementation of comprehensive multi-objective models is new and unexplored. Existing models have optimized, at most, three variables (commonly referred to as objectives) at a time [53], [54]. Such models do not consider embodied impacts, such as embodied carbon and embodied energy, in conjunction with durability and performance objectives, such as chloride permeability, strength, cost, and workability. Work by Liu, et al., [55] conducted within a factorial design of experiments is one of the most holistic attempts at mixture design optimization, as it considered optimizing five objectives: strength, freeze thaw durability, embodied energy, embodied carbon, and chloride durability. However, due to the limited capabilities of experimental factorial design, only nine optimal concrete mixtures were produced, and no useful relationships were provided to aid in the development of addition optimal mixes without subsequent experimentation.

Although there have been attempts at designing optimal concrete mixtures considering multiple criteria, existing methods, models, and mixes are either (a) not holistic in their approach or (b) limited in their applicable scope and thus not useful beyond a very limited application or data set. Previous work on multi-objective optimal mixture design models, however, indicates the potential and desire for such work. By taking a holistic and unified approach this work aims to improve upon

previous concrete mixture design optimization methods and proposes an alternative, holistic framework to traditional mixture design methodologies.

3.1.1. Scope of work

To address the limitations of the previously proposed methodologies for multi-objective optimization of concrete mixtures, this chapter presents a new methodology that combines multi-objective evolutionary algorithm (MOEA)-based search with a set of property prediction models, namely existing empirical and numerical relationships that link concrete mixture proportions to strength, workability, chloride resistance, freeze-thaw resistance, and sequesterable carbon dioxide. The model, which operates within the Borg MOEA framework [56], is implemented in Python and follows a similar formulation to prior Borg applications [57], [58]. In a new application, Borg is demonstrated herein to produce a suite of optimized mixture design solutions that elucidate tradeoffs of competing objectives.

The model defines six objective functions for strength, workability, cover depth (a chloride-induced corrosion durability parameter), embodied energy, embodied carbon, and cost that are used by the MOEA to evaluate optimal solution sets. Calibration of the modeling approach was performed by comparing results to industry data in order to ensure that solutions from the model are reasonable and realistic. Chloride transport is modeled via implementation of either a simplified error function solution [38] or the 1D numerical finite difference solution, as discussed in **Chapter 2**. Concrete mixture design solution sets are produced for three case study applications, namely a cubic meter of concrete, a tilt-up wall, and a column, which elucidate, for the first time, the complex tradeoffs that underpin the design and proportioning of concrete mixtures.

3.2. LITURATURE REVIEW

A literature review was conducted to understand the current state of concrete mixture design as well as the history and progress made in concrete mixture design optimization. First, in **Section 3.2.1,** the current state and limitations of traditional mix design methodology are discussed. Subsequently, in **Section 3.2.2**, previous studies on optimal mix design are investigated. These include experimental (**3.2.2.1**), analytical (**3.2.2.2**), statistical (**3.2.2.3**) and semi-experimental (**3.2.2.4**) methods and their progress toward a holistic multi-objective concrete mixture design methodology.

3.2.1. Concrete mixture design

Conventional concrete mixture design first establishes performance requirements, which typically include strength, durability, fresh-state properties (i.e., workability) and any special requirements based on application and design (i.e., recycled content). The proportions of constituents (i.e., cement, fine aggregate, coarse aggregate, water) are subsequently chosen to ensure required performance. Workability is key for fresh-state performance, as it governs the mixture's ability to be pumped, placed, and compacted. Similarly, compressive strength is often of primary concern for hardened-state performance of concrete mixtures [59]. While existing mixture design methodologies are different in how they approach informing their mixture design procedure, all utilize empirically derived relationships, charts, and graphs experiments. Economic cost is not addressed directly in current mixture design methodologies, but rather through the performance requirements and capability of the concrete batch facility to source constituents at a desired price. Although concrete is known to have significant environmental impacts [1], [60], [61], primarily as a result of cement production, existing mixture design methods do not consider detailed cost accounting of the energy or carbon of a mixture. This

section discusses the current state of concrete mixture design in the United States, as well as existing studies and methods for designing optimal concrete mixtures.

3.2.1.1. Current State

The first unified method of mixture design was established by the American Concrete Institute (ACI) Committee – 211 [62]. This method has helped to inform all other methods, such as the British Mixture design method and Bureau of Indian Standards method [63]. To this day in the United States, the primary guide is still produced by the Portland Cement Association (PCA) and ACI and constitutes two main methods. The first method, a prescriptive approach, prescribes allowable ranges of constituent proportions that can achieve set performance goals and takes two forms. One form is based on the estimation of a desired concrete weight per unit volume while the second form is based on calculation of an absolute volume of concrete ingredients (often called the absolute volume method, AVM). Conversely the second method, a performance-based approach, allows the designer to propose any proportioning that meets the performance goals. This approach requires experimental testing to provide data that ensure performance goals are met. Either method often relies on a time-intensive, trial-and-error approach, where mixtures must be tested following design and re-design, as necessary. This bottom-up approach leads to the design of a single acceptable mixture, rather than a truly optimal one that accounts for multiple design considerations. Throughout the past several decades. this lengthy, iterative methodology that governs the design and control of concrete mixtures in the United States has remained fundamentally unchanged [64], [65].

3.2.1.2. Limitations of Existing Models

As previously mentioned, most countries can trace their mixture design standards back to the ACI methodology. This common heritage has resulted in a

uniform and unchanged landscape of concrete mixture design. Thus, all standards and countries suffer from a similar problem when designing concrete mixtures – the inability to efficiently optimize multiple, often competing, performance objectives. The AVM, for example, has no provisions to advise increases in strength beyond what was prescribed by the structural engineer to ensure sufficient durability to chloride ingress and carbonation. Similarly, little information is provided in the mixture design methodologies to tie SCM addition to durability, strength, or workability. In general, a low water-to-cement ratio (*w/c)* acts as the indicator of strength and durability, while workability is tied to air content, aggregate gradation, and admixture additions [65]. Although relationships for many of these behaviors still need to be refined, current methods do not quantify or evaluate tradeoffs between competing properties (i.e., air content, strength). Rather, current methodologies prescribe bounds on mixture-component content, within which no problems are expected to arise for the designer (i.e., minimum w/c ratio, minimum air content).

Although the accounting of embodied impacts (i.e., environmental costs) for constituents is a relatively new science [66], carbon and energy accounting is beginning to affect concrete mixture design. Given that concrete production is one of the most energy and resource-intensive processes in construction and that concrete is the second most consumed constituent on Earth [1], it is increasingly vital for modern concrete mixture design methods to take environmental impacts into account. Environmental impacts in this study are measured in primary energy use [MJ/kg] and global warming potential in equivalent carbon-dioxide [$kgCO_{2e}$/kg]. However, other environmental metrics and indicators, such as eutrophication potential, ozone depletion potential, smog potential and acidification potential,

specified by ISO standards [66], are increasingly used to evaluate environmental performances of products and processes.

Traditional mixture design methodologies do little to address innovative and high performance mixture additives or design criteria. For example, no guidance is provided for including impacts of recycled aggregate (of any type) in the mixture process by ACI. Similar issues arise when using various admixtures not covered by the narrow selection that ACI includes in their mixture design process. Current methods also lack capability for designing new and increasingly prevalent forms of concrete. Mixtures that are designed to be self-consolidating or high-performance exist outside the data set used to inform the ACI design methods and, thus, cannot utilize the ACI prescriptive procedure in their mixture formulation. Rather, they must be designed following the performance-based, trial and error approach. These mixtures are designed through experimental optimization, since such computational optimization is not compatible with the linear and highly iterative process of the ACI method.

3.2.2. Concrete mixture design optimization methods

Although methods like those proposed by ACI and PCA are often used to design concrete mixtures, there are other tools that exist to aid design. Academia and industry have proposed a variety of amendments to (and replacements for) methods like the AVM [54], [67], [68], in acknowledgment of the limitations discussed previously. The literature reviewed in this section explicitly focuses on optimization techniques that have been proposed to best inform this work on designing concrete mixtures using a many-objective optimization approach.

In general, concrete mixture design optimization methods can be separated into four classifications [69]: experimental, analytical, statistical and semi-experimental. Each is further discussed in more detail.

3.2.2.1. Experimental Methods

Experimental, or parametric, methods involve an extensive series of experimental tests. Mixture designs that are formulated using this parametric analysis approach are dependent on two main parameters: (1) the size and depth of the investigated sample size (number of cylinders tested, parameters varied, mixture design goal variation) and (2) the type, form, and availability of local constituents used in the mixture [70], [71]. Improper design of these parameters might cause the performance characterizations of these mixtures to not be generalizable to other design scenarios. Moreover, even though this method is time- and constituent-intensive it often leads to a limited set of optimal mixture designs [72], [73].

3.2.2.2. Analytical

Analytical optimization methods can be very efficient at modeling phenomena but are often complicated to derive and can oversimplify the behavior of a system. Analytical methods for application in concrete mixture design are based on fundamental elements of math (i.e., geometry) that aim to predict a mixture parameter or design a whole mixture without relying on existing data or relationships. Analytical methods often help in searching for an optimum concrete mixture. However, no single analytical model currently exists to generally optimize a concrete mixture. The intent of these methods is to cut down on the time required to investigate certain constituents or the whole design of a mixture by fully defining it. The particle packing method proposed by Golterman, et al., [74] is an example of an analytical concrete optimization method. The method optimizes the type, size, quantity, and gradation of aggregates in a mixture. Using detailed knowledge of both aggregate and cement particle size, specific weight, and mathematical relationships describing the distribution or "packing" of circular particles, an

optimal concrete mixture for strength and cement content can be designed via optimization of aggregate gradation. The optimization is achieved through a geometric modeling procedure which can be either prescriptive or numerical [75]. In the latter case, a computer model governed by geometric relationships alone can find the optimal distribution of aggregate size that results in a minimal amount of void space. By minimizing void space, the particle packing method designs a mixture with the lowest possible paste ratio, while improving strength compared to similar (w/c) mixtures [59]. Following optimal packing design, experimental optimization can be used to investigate and optimize remaining properties of interest. The optimal packing ratio can serve as a more informed starting point for experimental study and can be varied to support additional concrete functionality (i.e., inclusion of SCMs).

3.2.2.3. Statistical

Statistical methods rely on existing experimental data to inform optimal mixture design. In general, these methods consist of taking in large numbers of existing mixture design data that through regression analyses are used to define curves. Regression analysis most commonly takes the form of a multi-linear regression where all variables are assumed to be independently and linearly related to the properties being optimized. Typically, response surface methodologies are used to plot and visualize the tradeoffs and impacts of the various decision variables [75]–[80]. Additionally, multi-variable empirical relationships can be defined to help predict performance in alternative design methods. Such a technique has been used to optimize as many as five mixture parameters based on eight variables [80]. However, multiple regression analysis can be a poor prediction method when used on highly correlated variables, as is often the case for concrete (i.e. strength and durability). Additionally, utilizing linear regression methods within a multiple

regression framework can lead to overly simplified non-linear trends and again, poor predictions of variables. For these reasons, statistical concrete optimization models have been unable to optimize more than two variables simultaneously [76], but can be utilized to individually optimize more than two variables given the same data set.

Use of a statistical method to produce an optimal mixture design is limited to the range and resolution of the experimental data used to inform the model. To help combat this limitation, several statistical mixture design models have employed Monte Carlo or augmented factorial design method to further increase the data range and resolution over that of the obtained experimental data [77], [78]. The same statistical tools can be used to create more complete response surface plots and empirical relationships. Response surfaces are made of up to a second order response regression function as seen in Muthukumar [79], where each regression can optimize for a single objective variable based on all independent variables. Using a desirability function, these responses can be combined to optimize the problem based on designer specified weights whose values are highest for what the designer considers the important performance objectives (i.e. strength) [81].

Current statistical concrete optimization approaches often focus on optimizing a mixture for a single application or set of constraints. These approaches include optimization of concrete using blends of metakaolin and fly ash [81], optimizing fly ash content in concrete [82], and optimizing admixture doses in pervious concrete [76]. Statistical model approaches have generated results that illustrate many of the trade-offs in concrete mixture formulation. Additionally, through tools such as multiple-regression analysis, statistical concrete optimization research has led to many concrete property prediction relationships that can be utilized by semi-experimental optimization methods.

3.2.2.4. Semi-Experimental

Semi-experimental methods are a bridge between analytical methods and statistical methods. These methods combine empirical prediction models and experimental data sets with analytical techniques, such as artificial neural networks (ANN), genetic algorithms (GA), harmony search (HS), and others. In this set of methods, it is important to distinguish the prediction methods from those that can be used to optimize. Increasingly, ANNs have been used as prediction methods for various concrete properties [83]–[85]. The advantage of ANN over empirical relationships is that relationships between large numbers of predictor variables can be found without becoming too complex or suffering from over-fitting issues. However, ANNs do require large and diverse population sets to train them so they can predict properties over a large range of predictor variables [86]. Meanwhile, GA and HS are emerging tools for mixture design optimization that perform beyond existing, primarily statistical methods, through their use of a global optimization search.

One of the first applications of one of these techniques to concrete mixture design was performed by Lim, who utilized a genetic algorithm to optimize the prediction of strength and slump of a concrete mixture with respect to seven decision variables [87]. The GA was informed by multiple regression models derived from a set of 181 experimentally tested mixtures with the final algorithm being compared against four of the before-mentioned mixtures to inform the model error. This error was then optimized further with the GA by increasing the algorithm's population size.

Following this early work many studies were conducted on alternative prediction techniques, namely neural networks, in combination with statistical optimization techniques [68], [88]–[90]. Additionally, Cheng, et al., [91] recently

attempted accurate prediction of mixture design parameters using a GA with an evolutionary support vector machine. In this work, the model was informed by over 1030 sample mixture data points and focused on accurate prediction of strength and cost with consideration of eight decision variables.

Although studies have attempted a comprehensive optimization of a mixture design for a single objective [59], [71], [75], [77], [79], [92]–[96] many have not moved beyond comprehensive characterization of a single objective to address multiple objectives. Nonetheless, one of the first studies to optimize a mixture design in a global solution space was conducted using a genetic algorithm [95]. The study focused on optimizing mixture strength with decision variables being cement content, fly ash, water, sand, coarse aggregate, and superplasticizer. An existing empirical equation that related all of the decision variables to the objective was utilized as the objective function. Additionally, constraints on ratios of constituents were provided based on statistical analysis of experimental data. The result of this study was a set of five optimal mixtures for five various strength targets. Similar studies have used either GA or HS to optimize single-objective mixtures, but never with more than eight decision variables or for objectives other than strength and cost [96].

A multi-objective optimization approach was proposed by Lee [97] in 2012, which involved a HS algorithm. HS is a meta-heuristic algorithm that is modeled after theories in music and aims to find the perfect harmony of solutions given a problem formulation. The algorithm was used to optimize both strength and slump with regard to six decision parameters. Regressions used in [87] were utilized to correlate objectives and decisions to make the model highly comparable to the previous GA proposed by Lim [87]. The predictive capabilities of such a model were

tested against both a genetic algorithm and ANN and was found to be superior, concluding that optimization would be a more accurate modeling approach [97].

Although modern computational techniques such as ANN, GA, and HS have been found to accurately predict and optimize concrete mixtures, they have, so far, only been implemented on a small scale. The only optimizations beyond 2 objective variables are still statistically based. Methods utilizing more computationally intensive and global search-oriented approaches have only been applied to very specific mixture formulations and often do not include durability or embodied impacts with the exception of Kim, et al., [54].

A contributing factor to the lack of design consideration for embodied impacts, such as carbon and energy, is the relatively new, standardized methodology for accounting for such impacts. Constituents used in concrete are known to contribute to global CO_2 production, energy use, and raw constituent depletion. However, the magnitude is not well understood. Organizations such as the International Standards Organization (ISO) and US Green Building Council (USGBC) have defined and encouraged the proliferation of environmental accounting throughout the building industry. Accounting is done through a process known as a life cycle assessment (LCA) and, for building materials and products, is commonly reported in terms of embodied energy and embodied carbon—two environmental metrics of interest in the work presented herein.

3.3. METHODOLOGY

3.3.1. Optimization approach

For design optimization, this study utilizes a multi-objective evolutionary algorithm (MOEA). As previously discussed, concrete mixture design can be

holistically approached as a multi-objective design problem. A series of empirical and numerical relationships that predict fresh-state and hardened-state concrete properties of interest are embedded within the search process of the MOEA to evaluate the performance of candidate mixture designs—solutions to the optimization problem (multi-objective optimization problems by definition have multiple potential solutions). Through this integration of simulation models with the MOEA, the most complete models can be used, which provides high confidence in the solution set. Given the correlated and multi-variable nature of many concrete properties, MOEAs are a desirable optimization approach, particularly when considering large numbers of objectives and decisions.

Multi-objective optimizations output a set of solutions known as the Pareto-optimal set. Solutions are informed by models of the desired performance objectives whose inputs, known as decision variables, are used to calculate the value of the performance objects. A mathematical concept called 'non-domination' is used to define, based on a set of performance objectives, which solutions remain in the set. A solution is non-dominated if its performance in all objectives is not exceeded by any other feasible solution in all objectives. In other words, non-domination is a way to elucidate tradeoffs between different objective functions of the design problem within the realm of optimal solutions. For a two-objective minimization problem, the non-dominated set is the smallest values of the first objective that can be obtained for every level of the second objective. The term 'Pareto-optimal' indicates the best non-dominated set among all feasible solutions. Note, however, that finding the true Pareto-optimal solution set for a given problem is often computationally difficult, because it requires enumeration of every possible feasible solution to the problem. Therefore, studies using MOEAs find a set of solutions termed the Pareto-

approximate set, which is the best-known approximation of the true Pareto optimal [98].

This study aims to explore the many tradeoffs expected in concrete mixture design, while providing insight on potentially novel concrete mixture designs. In this study, models where chosen that allow for the variation of seven design parameters, or decisions, as summarized in **Table 3-1**. Six objectives are calculated using these decisions and are reported in **Table 3-2,** along with which decisions impact each objective. Additionally, two fixed design criteria are utilized – design service life and chloride exposure – which constrain the solution set to those mixtures that can meet the duration and exposure criteria based on the durability modeling.

Table 3-1. Decision variables that inform the concrete mixture design

Decision	Lower Limit	Upper Limit	Description
cement	550	700	Cement Content (kg/m^3)
wc	0.25	0.75	Water to Cement ratio
air	1	6	Air Content (% of volume)
ca	55	75	Coarse aggregate ratio (% of volume)
scm	0.5	4.49	SCM variable rounded to be 1, 2, 3 or 4
replace	0	30	SCM replacement (% by weight of cement)
rca	0	1	Recycled agg. replacement (% by weight of ca)

Table 3-2. Objective functions and those decisions that impact them

Objective	Description	Affected by
Strength	Maximize compressive strength (psi) at 28 days	cement, wc, air
Cost	Minimize cost ($) of mixture constituents per functional unit	cement, ca, scm, replace, rca
Cover	Minimize cover (m) needed to achieve a prescribed-fixed service life	wc, scm, replace, rca
Workability	Maximize slump (cm) of fresh state mixture	cement, ca
Embodied Energy	Minimize energy (MJ) per functional unit	cement, ca, scm, replace, rca
Embodied Carbon	Minimize Global Warming Potential (kgCO2e) per functional unit	cement, ca, scm, replace, rca

3.3.2. Borg multi-objective evolutionary algorithm

Although MOEAs provide the potential of finding a high-quality Pareto optimal solution set, they are not always guaranteed to do so in a reasonable number of function evaluations. Given the limited use of MOEAs in concrete mixture design optimization and the ambition of this work to model multiple concrete mixture variables, avoiding potential failure of the MOEA algorithm was critical. Kasprzyk, et al., [58] and Reed, et al., [45] define the major failure modes of MOEAs as being related either to mathematical structure (over complex problem formulation) or search limiting user-chosen algorithm operators. With the goal of limiting the potential of these failures, the MOEA used in this study is the Borg MOEA [56]. It was chosen because of its capabilities as an auto-adaptive algorithm

which has led to superior performance relative to other state-of-the-art MOEAs [99], [100]. Auto-adaptivity means that Borg can combine existing MOEA algorithms and their search operators to avoid algorithm failure. Prior studies of algorithm performance such as [58], [99] have shown that these design characteristics help Borg exhibit effective performance on a wide variety of problems. .

Borg also utilizes epsilon dominance [101], which allows the user to define the precision of each objective. Epsilons act to measure search progress and to ensure solution diversity and, thus, require meaningful choice based off understanding of the range over which objective values may vary [57]. In addition to tailored use of search operators, Borg is also adaptive in its use of population sizing [102]. As an additional means of ensuring adequate solution diversity, the Borg will inject new and diverse populations into the simulation set in an effort to fully explore the decision space.

For this study the only user-controlled parameters of the Borg MOEA were the epsilon values and total number of function evaluations. Epsilons were determined based on the lowest desired precision and the assumed range of the objective function solutions. Applicable ranges vary based on the type of investigation, but, in general, epsilons were set per **Table 3-3**. Epsilons could be redefined after running the model, since objective ranges could be larger than expected and the solution set size was larger than desired. A similar epsilon tailoring approach was employed by Kollat and Reed [102]. In general, higher epsilon values lead to a reduction in solution set size. The operation of the Borg in this study is implemented with its default parameters [56] summarized in **Table 3-4**. The only user-chosen parameter was the number of function evaluations, which was set to 100,000 following preliminary investigations that indicated that further

increase in function evaluations would result in negligible differences in solution performance.

Table 3-3. Default epsilon inputs for the model

Objective	Chosen Epsilon
Cost	1
EE	5
Strength	100
Cover	0.005
EC	1
Workability	1

Table 3-4 Borg parameters and values used

Parameter	Chosen	Default
Initial Population Size		100
Maximum evaluations	100,000	N/A
Injection rate		0.3
SBX rate		1.0
SBX distribution index		15.0
PM rate		$1n_{decvar}$
PM distribution index		20.0
DE crossover rate		0.1
DE step size		0.5
UM rate		$1n_{decvar}$
PCX number of parents		3.0
PCX number of offspring		2.0
PCX eta		0.1
PCX zeta		0.1
UNDX number of parents		3.0
UNDX number of offspring		2.0
UNDX eta		0.5
UNDX zeta		0.4
SPX number of parents		3.0
SPX number of offspring		2.0
SPX epsilon		0.5

3.4. MODEL DEVELOPMENT

The relationships used to inform the objective functions used by the MOEA are presented in this section. ACIs absolute volume method (AVM) is utilized as the primary driver for determining mixture proportions. Empirical relationships from previous research are then implemented to inform the impact of the AVM mixture proportioning on the objectives of strength, slump, and, given an exposure, required cover depth. Additionally, the 1D numerical service life model presented in Chapter 2 is utilized to provide a more comprehensive durability prediction capability. The specifics of the LCA conducted within each mixture design are discussed, along with (a) a novel methodology to account for sequesterable CO_2 and (b) the default life cycle inventory (LCI) values used to calculated environmental impacts.

3.4.1. Mixture proportioning by absolute volume method

Given its ubiquity and ease of implementation, the ACI absolute volume method (AVM) was chosen as the technique for developing the mixture design proportioning in this study. Although alternative methods such as particle packing methodology could have been used, the intent of this study was to show a proof of concept to use MOEA to design concrete mixtures within a flexible framework that could add more mixture design steps beyond AVM in future work. None of the empirical relationships or tabulated values for the design variables commonly used in conjunction with the AVM was employed in this model. In contrast, the MOEA framework first selects values of cement content, w/c ratio and ca ratio that would then be used to inform the AVM proportioning. Then, the empirical relationships and numerical models presented in this section were utilized to include those properties traditionally determined from tables in the AVM. In general, the procedure of the AVM in this method takes the form illustrated in **Figure 3-1**. Inputs are given as single values chosen from the ranges in **Table 3-1**. The outputs

of this process form the mixture design for a single solution. They can be further converted to weights needed to make a cubic yard of said concrete mixture.

All inputs to the absolute volume method are determined by the Borg MOEA algorithm and are within the bounds given to it as decision variables. After the absolute volume calculation is completed, the volume of each mixture constituent is then converted into a kg per cubic meter (kg/m^3) and lb per cubic yard (lb/yd^3) of constituent to constitute the final mixture design. Both units are used because the various empirical relationships and numerical models employed where developed in both imperial and metric units.

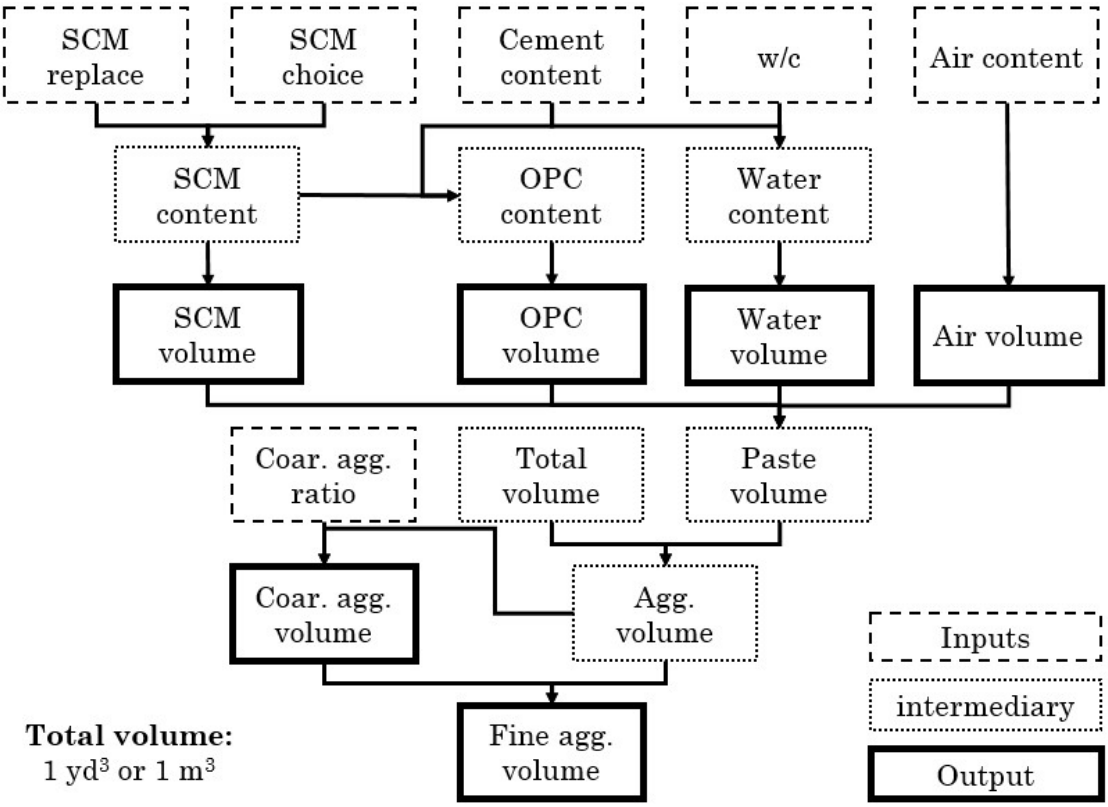

Figure 3-1. Flow chart of AVM implementation in the model.

3.4.2. Empirical property prediction relationships

As discussed, objectives are a function of the decision variables, whose values are chosen by the optimization algorithm in conjunction with the process of the absolute volume method. Equations and models of constituent properties, further referred to as prediction relationships, link the two and are what determine the precision and accuracy of the objectives for a given set of decisions. Without well-constructed and validated prediction relationships, objective tradeoffs would likely not be representative of real world behavior. No perfect prediction relationships exist, but all those currently proposed have the potential to serve a function within this model if found to be of adequate quality. The following section presents the selection and definition of the objective prediction relationships utilized within this study. Empirical relationships are discussed that were used to predict (1) strength, (2) workability, (3) chloride diffusion via a simplified model, and (4) SCM addition impacts on chloride diffusion.

3.4.2.1. Strength

Previous research has shown strong correlations between w/c ratio, cement content, air content, and strength [103]–[105]. While other variables including SCM additions for fly ash and silica fume have also been shown to impact strength, fewer and less accurate prediction equations exist [104]. The relationship utilized in this study to predict mixture strength is a model proposed by Popovic [103]:

$$f_c = \frac{51{,}290}{23.66^{w/c+0.000378 \cdot c+0.0279a}} \qquad \textbf{Eq. 3-1}$$

Where f_c is the predicted 28-day strength, w/c is the water-to-cement ratio, c is the cement content (lb/yd^3) and a is the air content (%). When compared to recent, more

complex relationships, this prediction relationship provided more accurate strength predictions based on industry data [104], [106].

3.4.2.2. Workability

Workability can be measured in a variety of ways including slump, flowability, and pumpability. A vast amount of work has been devoted to predicting the rheological properties of fresh-state concrete [107]–[110]. Many of these investigations involve artificial neutral networks (ANN), which are not trivial to implement nor does this approach output an easy-to-use relationship [111]. Well-documented regression-based empirical relationships exist to supplement the solutions from ANNs [83], [112]. The model used in this study is a modified form of that proposed by Hoang et al., [112] derived using the Least Squares Estimation method. The model considers impacts from all four of the main mixture constituents (i.e., cement, water, coarse aggregate, sand) as well as superplasticizer dosage:

$$y = 36.22 - 12.47x_1 - 27.03x_2 - 24.56x_3 - 7.39x_4 - 3.00x_5 - 1.18x_6 \quad \textbf{Eq. 3-2}$$

where y is the slump of the concrete (cm), $x_1, x_2, x_3, x_4, x_5, x_6$ are the normalized contents of cement, natural sand, crushed sand, coarse aggregate, water and superplasticizer respectively. Normalization of the inputs is performed with respect to the data statistics (maximum and minimum) provided by Hoang with the development of the model. Normalization of input variables was necessary because Eq. 3-2 was specifically formulated to accept only normalized values of each input variable. Crushed sand and superplasticizer are not considered in the model formulation of this study and were thus ignored, leading to the modified version accepting only x_1, x_2, x_4 and x_5 as inputs.

3.4.2.3. Simplified Chloride Diffusion

A desire to include sophisticated chloride transport as an objective function within the optimization was countered by the known computational complexity of numerical approaches, such as those covered in Chapter 2. A simplified approach for calculating chloride concentration through a media with time was necessary to include within the model. Although a more precise, complete and computationally expensive model has been included in the model, the value of a computationally quick, approximation was such that it too was included in the model for use when accuracy in durability prediction was of secondary concern. A common, well-known steady-state solution to Fick's Second Law of Diffusion for use in predicting chloride concentration through a media with time is the error function solution [38]:

$$c(x,t) = c_0\left(1 - erf\frac{x}{2\sqrt{D_c t}}\right) \qquad \textbf{Eq. 3-3}$$

where *c(x,t)* is the chloride concentration at a distance *x* from the concrete surface at time t (service life in this model), c_0 is the chloride boundary condition (a fixed user input for all solutions) and D_c is the apparent diffusion coefficient for the concrete. In the simplified case, apparent diffusion coefficient is taken to be the value of the diffusion coefficient at 28 days regardless of the service life of the concrete.

3.4.2.4. SCM impact on diffusion coefficient

The apparent diffusion coefficient used in Eq. 3-3 is affected by w/c ratio and SCM addition according to the following:

$$D_{28} = 2.17 \times 10^{-12} e^{(w/c)/0.279} \quad \text{Eq. 3-4}$$

$$D_{SF} = D_{28} \cdot (0.206 + 0.794 e^{(-SF/2.51)}) \quad \text{Eq. 3-5}$$

$$D_{UFFA} = D_{28} \cdot (0.170 + 0.829 e^{(-UFFA/6.07)}) \quad \text{Eq. 3-6}$$

$$D_{MK} = D_{28} \cdot (0.191 + 0.809 e^{(-MK/6.12)}) \quad \text{Eq. 3-7}$$

where w/c is the water-to-cement ratio, *SF, UFFA* and *MK* are the replacement percentages of silica fume, fly ash and metakaolin respectively and D_{28}, D_{SF}, D_{UFFA} and D_{MK} are diffusion coefficients at 28 days for OPC silica fume, fly ash, and metakaolin. The 28 day diffusion coefficient modification for slag replacement was found by Riding, et al., [10] to be negligible, resulting in slag cement mixtures having the same 28-day diffusion coefficients as OPC when Eq. 3-3 is implemented.

3.4.2.5. 1D diffusion model

In addition to the simplified diffusion model in **3.4.2.3** a numerical 1D diffusion model with greater capability at addressing chloride concentration prediction was included. This was done to improve the model's accuracy when durability prediction was of primary concern but comes at the cost of computational speed. The 1D diffusion model utilized in this study is based on that proposed and validated in Chapter 2. There are two primary differences between the model used for mixture design optimization and that used in Chapter **2**. First, this model is of a deterministic form, requiring fixed inputs that are outlined in **Table 3-5**. Second, within the optimization model, service life is taken to be a user input. Therefore, the 1D diffusion model was used to solve for a required cover depth given a chloride exposure condition rather than chloride concentration at a given lifetime.

Table 3-5. Deterministic 1D diffusion model parameters

Service-life modeling parameter	Value	Units	Refs (Chapter 2)
Time-to-corrosion-initiation			
Recycled aggregate diffusion coefficient, D_a	12.5	kg/m^3	[17], [20]
Chloride threshold, c_t	0.7	kg/m^3	[39]–[41]
Time to corrosion cracking			
Tensile strength, f'_t	3.75	MPa	[11]
Modulus of elasticity, E	30	GPa	[11]
Phi (creep coefficient), ϕ	2	-	[33]
Poisson's ratio, v	0.18	-	[33], [42]
Density of rust, ρ_r	3600	kg/m^3	[33], [43]
Density of steel, ρ_s	7850	kg/m^3	[33], [43]
Thickness of porous region, t_p	12.5	µm	[33], [44]
Corrosion rate, i_{corr}	2.5	$\mu A/cm^3$	[33], [43]
Alpha, α	0.622	-	[33]
Mild steel rebar diameter, d_b	9.5	mm	

3.4.2.6. Recycled aggregate modeling

As with the model presented in Chapter 2, the application of the 1D model within this study includes a capability to model the effect of recycled aggregates. The placement of aggregates was made random within the entire depth of concrete. This was an improvement over the model utilized in **Chapter 2**, as it allowed

contaminated aggregates to be placed on either side of the reinforcing steel. Additionally, because failure was prescribed to occur at a specific time rather than concentration, aggregates could be placed within the whole cover. Aggregate placement was random and ensured no overlap. The same recycled aggregate diffusion coefficient employed in **Chapter 2** was used herein, continuing the assumption that recycled mortar aggregates are used for mixture proportioning.

3.4.3. Life Cycle and Life Cycle Cost Accounting

The LCA conducted within the MOEA framework quantifies the embodied energy and embodied carbon of resulting mixture designs for each case study application. The environmental accounting is based on constituents chosen by the designer and proportions specified by the MOEA algorithm. In addition to the LCA outlined in this section, the economic cost accounting, further referred to as the life cycle cost accounting (LCCA), pertaining to the same goal and scope as the LCA, is discussed.

All constituents used in this study are considered to be virgin unless otherwise noted (as with recycled aggregate). No constituents were considered to be byproducts, coproducts, or waste unless otherwise noted. All impacts are taken to be attributional and, thus, pertain only to the current state of conditions (manufacturing, economy, etc.) at the time of the study. The impacts metrics considered in the model are embodied energy (MJ), embodied carbon ($kgCO_2e$), and cost ($). Impacts for the LCA and LCCA were calculated according to the following equations:

$$EE = \sum_{c} m_c * EEC_c \qquad \textbf{Eq. 3-8}$$

where, *EE* is the embodied energy of a constituent, m_c is the constituent's mass and EEC_c is the Embodied energy coefficient specific to that constituent displayed in **Table 3-6**. Similarly, the impacts of embodied carbon and cost can be calculated as follows:

$$EC = \sum_{c} m_c * ECC_c \qquad \textbf{Eq. 3-9}$$

$$Cost = \sum_{c} m_c * CostC_c \qquad \textbf{Eq. 3-10}$$

where *EC* and *Cost* are the impact embodied carbon and cost respectively, m_c is the constituent mass and ECC_c and $CostC_c$ are the embodied energy and cost coefficients for the constituent respectively. Transportation impacts were calculated by a similar method to the constituent impacts above but required the addition product of transportation distance and generally took the following form:

$$TransImpact = \sum_{c} m_c * TransImpactCoeff_c * d_c \qquad \textbf{Eq. 3-11}$$

where *TransImpact* is either the embodied energy, carbon or cost, m_c is the mass of the constituent being transported, $TransImpactCoeff_c$ is the transportation method impact coefficient for either embodied energy, carbon or cost and d_c is the distance the constituent traveled. The coefficients utilized in this equation for both a truck and train transportation method are displayed in **Table 3-6.**

3.4.3.1. Life Cycle Inventory

Data collection for the inventory analysis came from a variety of sources. In general, data was derived via research and industry data varying from

manufacturer conducted LCAs and EPDs, to national company averages. Whenever possible, data was found that pertained to the United States. Researched and default values of impact coefficients for the life cycle accounting done in this study are presented in **Table 3-6.** These values are the default values for all model runs in this study. The transportation data is presented for both a truck and train, with the truck being considered the default transportation method. All data was considered to be complete based on its origin in scholarly research or manufacturer EPD. Through comparison with industry data, consistency of the mixture design tool to real-life data was ensured and is further discussed in **3.6.1**.

Table 3-6. Life cycle inventory coefficients used in this study

Constituent	EEC_C (MJ/kg)	ECC_C ($kgCO_2e$/kg)	$Cost C_c$ ($/kg)	Refs.
Cement	5.9	0.9	0.17	[65], [113]–[115]
Fly Ash	0.1	0.01	0.065	[116], [117]
Slag	1.6	0.146	0.106	[118], [119]
Silica Fume	0.04	0.7	0.44	[113], [120]
Metakaolin	2.08	0.6	0.36	[121]
Coarse Aggregate	0.1	0.0061	0.021	[122], [123]
Recycled Aggregate	0.1	0.003	0.01	[124], [125]
Fine Aggregate	0.08	0.0076	0.025	[122], [123]
Water	0.01	0.0001	0.005	[113]
Transportation	(MJ/kg-mi)	($kgCO_2e$/kg-mi)	($/kg-mi)	
Truck	0.0039	0.00115	0.005	[126]–[128]
Train	0.00034	0.00032	0.0009	[126], [127], [129]

Transportation distances are well known to exhibit dominant impacts on the embodied energy and embodied carbon of a constituent. Each constituent utilized in the model was assigned a transportation variable whose value was informed by industry and literature research, reported in **Table 3-7**. Those constituents whose distance from a specific mixture location is highly regionally dependent, namely SCMs and recycled aggregates, were given representative values that can be changed when a precise mixing site and thus SCM sourcing distances are known.

Table 3-7. Constituent sourcing distances

Constituent	Distance (miles)	Source
Cement	50	[130], [131]
Fly Ash	100	--
Slag	100	--
Silica Fume	100	--
Metakaolin	100	--
Coarse Aggregate	50	[130], [131]
Recycled Aggregate	100	[124]
Fine Aggregate	75	[130], [131]
Water	20	[130], [131]

3.4.4. Case Study Functional Units

Three case study applications were considered in this work: a cubic meter, a tilt-up wall, and a column whose details are reported in **Table 3-8**. Each case study was chosen to help illustrate the power of the MOEA approach and elucidate

multiple tradeoffs between competing objectives. The cubic meter was defined as a unit volume of concrete mixture, chosen to be one cubic meter. The results from the unit volume analysis were used to calibrate modeling results with industry data. No quality or duration was given to the cubic meter, resulting in mixture durability being ignored in the model. Unlike the cubic meter unit, the tilt-up wall was defined as, given a chloride exposure, a 1 m x 1 m x cover depth volume of concrete necessary to resist corrosion for a set number of years. In the final case study application, the column was defined as the volume of concrete required to support a 1,000-kip load and, simultaneously, given a chloride exposure, resist corrosion for a set number of years. As an input parameter for the user to specify, service life can be any number of years. Therefore, for a given set of optimal mixtures, anticipated (design) service life of the model must be included as an input.

Table 3-8. Case study applications (functional units) for model implementation

Name	Quantity	Quality	Duration
Cubic meter	One cubic meter of concrete	None	None
Tilt-up wall	A meter by meter by cover concrete depth wall	Resist Corrosion	Varies
Column	12-foot-tall concrete column	Support 1000 kips and Resist corrosion	Varies

Given the nature of the problem formulation in this study, all concrete mixture components considered in the model are considered in this LCA. In general, given that the MOEA can select some constituents on its own, constituents included are cement, sand, coarse aggregates, water, and SCMs. For this study, a cradle to gate system boundary was assumed. The system boundary primarily considers

manufacturing (A1) of mixture constituents and transportation (A2) of mixture constituents to the batch plant. Transportation beyond the batching plant, where the mixture was made, is not included because it was considered to be independent of the mixture proportioning and manufacturing. Additionally, no stages beyond the batching plant are considered in this analysis with the exception of the inclusion of the use (B1) phase for the column case study, which can be turned on and off. Beyond this one exception, accounting for embodied impacts stops at the exit gate of the cement batch plant, prior to transportation to the construction site. Similarly, accounting for economic cost considers same the cradle to gate boundary with no considerations made for use-phase costs as they are too variable to estimate and do not universally correlate strongly with mixture design.

3.4.5. Modeling of sequestered carbon

Carbonation in concrete is often viewed only through the lens of durability [132]–[134] with concern placed on limiting the rate and quantity in concrete. Recently, carbonation has been shown to also be a non-trivial means of carbon dioxide sequestration in concrete members over their service life. Experimental papers have investigated the amount of carbon sequestered over 100 years [135] and the effects of blended cements on carbon sequestration [136]. A newly formulated model proposed by Souto-Martinez, et al., [137] utilizes existing relationships for carbonation depth as well as newly discovered experimental coefficients for various constituent impacts and carbonation potential. For implementation in this study, the calculated sequestered carbon for a solution mixture design was not set as its own objective as indicated by **Table 3-2** but rather was subtracted from the embodied carbon objective value.

The process of calculating sequestered carbon is discussed in three steps: the calculation of the carbon sequestration potential of the concrete paste, estimation of

the carbonation depth at the end of an object's service life, and combination of these two to predict the total sequestered carbon of an exposed (i.e., unpainted, unsealed) concrete object over its service life.

3.4.5.1. Sequestration potential of mixture paste

Using cement and carbonation chemistry, Souto-Martinez, et al., [137] first linked cement type and SCM type and replacement percentage to the carbonation potential of the cement paste in a concrete mixture:

$$C_m = \alpha - \beta \cdot y \qquad \textbf{Eq. 3-12}$$

where C_m is the mass percentage of sequesterable carbon per kg of carbonated cement paste, y is the percent replacement of SCM specified by the Borg algorithm and α and β are the cement and SCM carbon sequestration potential coefficients respectively. The coefficients for α and β are presented in **Table 3-9** and are a property of their respective constituents within the model.

Table 3-9. Cement and SCM coefficients for carbon sequestration potential

Cement Type	α	SCM Type	β
Type I	0.165	Fly Ash (Class F)	0.55
Type II	0.163	Fly Ash (Class C)	0.27
Type III	0.166	Slag	0.38
Type IV	0.135	Silica Fume	0.99
Type V	0.161	Metakaolin	0.55
White	0.203		

3.4.5.2. Carbonation depth

The model utilizes well known empirical relationship for predicting carbonation depth developed by the Portuguese National Laboratory [138]:

$$x = \sqrt{\frac{2ct}{R}} \cdot \left[\sqrt{k_0 k_1 k_2} \left(\frac{1}{t} \right)^n \right] \quad \textbf{Eq. 3-13}$$

where c is the environmental CO_2 concentration, t is exposure time (the service life parameter within this study), k_0 is equal to 3.0, k_1 and n are exposure specific parameters outlined in **Table 3-10**, k_2 is equal to 1.0 and R is the carbonation resistance coefficient calculated as:

$$R = 0.0016 \cdot f_c^{3.106} \text{ for cement Type I and II} \quad \textbf{Eq. 3-14}$$

and

$$R = 0.0018 \cdot f_c^{2.862} \text{ for cement Type III-V and White} \quad \textbf{Eq. 3-15}$$

For use in this study the exposure classification is specified as a user input when the sequestration model is called. The four exposure classifications proposed by Souto-Martinez, et al., [137] are also utilized in this model with XC1 indicating dry or permanently humid conditions, XC2 being humid, rarely dry, XC3 being moderately humid and XC4 being cyclically humid and dry exposure.

Table 3-10. Exposure classification for sequestered carbon modeling

Parameter	XC1	XC2	XC3	XC4
k_1	1	0.2	0.77	0.41
n	0	0.183	0.02	0.085

3.4.5.3. Total sequestered carbon

Carbonation was considered to act uniformly on all sides of the concrete elements considered in this study; however, this is not a requirement of the model as any geometry and configuration of boundary conditions can be accommodated by the model. Nonetheless, once the carbonation depth is calculated, the total volume of carbonated concrete can be computed:

$$V_c = SA \cdot x \qquad \textbf{Eq. 3-16}$$

where V_c is the carbonated volume, SA is the surface area of the concrete element and x is the previously calculated carbonation depth. It is noted that V_c cannot be larger than the volume of the object in question, so this constraint is mandated in the model implement for this study. From the total carbonated volume of concrete the mass of sequestered carbon within the object can be calculated as:

$$C_s = \phi_c C_m \cdot \left[V_c \cdot m\right] \qquad \textbf{Eq. 3-17}$$

where ϕ_c is the degree of carbonation (default 1.0) and m is the total cement content of the in the concrete mixture.

3.5. MODEL IMPLEMENTATION

Optimization was performed for nine different mixture design scenarios, each varying between one and four model inputs or properties as summarized in **Table 3-11**. One baseline Scenario is investigated for each of the three Cases: (1) a cubic meter of concrete, (2) a concrete tilt-up wall, and (3) a concrete column. Six Scenarios are investigated for (1) and (2), and two Scenarios are investigated for (3) in addition to the baseline. The Scenario numbers containing a 'C' pertain only to

the column functional unit. The choice of three case study applications was made to test the models ability to solve increasingly constrained problems.

Table 3-11. Scenario definitions for use with each functional unit (Case)

Scenario No.	Intent/context	Variables changed
0	Baseline model with no variable change	N/A
1	Coal power plants close and Fly Ash stock decreases	FA, $
2	Increase allocation accounting for Fly Ash impacts	FA, EEC, ECC
3	Recycled Agg. is close and abundant	Recycled Agg. distance
4	Distance for all SCM to appear in solution set	FA distance, SF distance, MK distance, SG distance
5	Transportation is by train rather than truck	Transport $, EEC, ECC
6	No admixtures are allowed so w/c must be increased to maintain workability	w/c lower bound
1C	Interior column that should not be larger than 16” x 16”	Functional unit area constraint
2C	Exterior bridge pier that should be larger than 36” x 36”	Functional unit area constraint

To establish a baseline for every functional unit, a default model, Scenario 0, was utilized, whose parameters are defined in **Table 3-12**. Additionally, the modified variables used throughout each scenario are also presented in **Table 3-12.**

Table 3-12. Value change in variables for Scenario 1-6

Variable of interest	Values		Scenario
	Scenario 0	Modified	
Fly ash $	0.065	0.44	1
Fly ash EE	0.1	2	2
Fly ash EC	0.01	0.2	
Recycled agg. distance	100	10	3
Fly ash distance	100	variable	4
Silica fume distance	100		
Slag distance	100		
Metakaolin distance	100		
w/c lower bound	0.25	0.35	6

Each functional unit requires unique parameters and inputs in addition to those presented in **Section 3.4** and in Table 3-12. The cubic meter, tilt-up wall, and column functional units are individually discussed, including further justification and information for their utilization.

3.5.1. Case 1: One Cubic Meter of Concrete

Unlike the subsequent two functional units, the cubic meter of concrete case has quantity, but no quality or duration. As a result, any structural load or hazard (i.e., chloride resistance) is neglected in the objective functions. All other objectives are calculated as outlined in **Section 3.4**. The cubic meter unit is investigated under each of seven scenarios, namely Scenario 0 (baseline) and Scenarios 1-6 outlined in **Table 3-11**. Results and discussion for the cubic meter investigations follow this section and begin in **Section 3.6.2**.

3.5.1.1. Inputs

Epsilons (see **Table 3-13**) were chosen that would reduce the solution set to a more manageable number for clearer display of Case 1.

Table 3-13. Epsilons for Case 1.

Objective	Chosen Epsilon
Cost	40
EE	100
Strength	100
Cover	0.005
EC	8
Workability	4

All objective function parameters and decision variables are based on the same inputs as specified in **Section 3.4.3**. Changes to the objective function parameters and decisions not covered in **Table 3-12** for Case 1 are shown in **Table 3-14**. Sequestered carbon and durability were not considered for the cubic meter unit and were thus turned off within the model.

Table 3-14. Distances for Case 1, Scenario 4.

Variable	Modified Value
Fly ash distance	300
Silica fume distance	150
Slag distance	100
Metakaolin distance	20

3.5.2. Case 2: Concrete Tilt-Up Wall

The tilt-up wall case study explored in this model is defined as a square meter in surface area by a variable depth of cover required to resist chloride-induced corrosion for a minimum specified design service life. Both the simplified and numerical 1D service life model was utilized. No accounting for sequestered carbon was included for this case study application. The tilt-up wall unit is investigated under each of the 7 scenarios, Scenario 0-6, as shown in **Table 3-11**. Results and discussion for the tilt-up wall investigations follow this section and begin in **Section 3.6.3**.

3.5.2.1. Inputs

The tilt-up wall, Case 2, utilized default epsilon values (**Table 3-3**). Implementation of the simplified and numerical 1D diffusion model utilized the exposure conditions and service life specified in Table 3-15. Initial recycled aggregate contamination for use with the 1D model is also included in **Table 3-15** and was required to be less than the corrosion initiation limit of the rebar (0.7) to prevent 100% aggregate replacement from resulting in an infinite, and thus unrealistic, cover depth. Additionally, all inputs for the numerical service life model were presented in **Table 3-5**. Exposure definitions for the numerical 1D model correspond to the classes specified in **Chapter** 2. The class can be specified as a user input integer from 1 to 5, with lower numbers being more extreme exposure conditions.

Table 3-15. Exposure condition variable values use in Case 2

Variable	Model			
	Simplified	units	Numerical 1D	units
Exposure	5	kg/m^2	3	Class
Service Life	25	years	25	years
Recycled agg. contamination	N/A	--	0.5	kg/m^2

3.5.3. Case 3: Concrete Column

Two different column types are considered in this study, as described in **Table 3-11**. The column unit is investigated under each of three scenarios, namely Scenario 0, 1C, and 2C, also outlined in **Table 3-11**. Results and discussion for the column investigations follow this section and begin in **Section 3.6.4**.

3.5.3.1. Inputs

Epsilon values for the column investigations in this study utilized the default values in **Table 3-3**. Employing the numerical 1D diffusion and carbon sequestration models required additional variables to be specified for the model per **Table 3-16**. The same values of aggregate contamination were used as specified in **Table 3-15**. Additionally, the cement type was specified as Type I, with application in exposure category XC1 for use in the carbon sequestration model.

Table 3-16. Additional inputs used for Case 3.

Variable	Column Type			units
	Baseline	Case 1C (interior)	Case 2C (exterior)	
Cl exposure	3	N/A	1	Class
Service life	100	100	100	years
CO_2 exposure	300	800	300	ppm
Load	1000	1000	1000	kips

3.6. RESULTS AND DISCUSSION

Results are presented and discussed in four parts. First, model calibration with industry data is discussed, along with a preliminary investigation into model sensitivity. Second, the results for Case 1 (**Section 3.6.2**), Case 2 (**Section 3.6.3**), and Case 3 (**Section 3.6.4**) are presented with respect to each scenario as outlined in **Table 3-11**.

3.6.1. Model calibration with industry data

First, the model was calibrated using a comparison with industry-acquired data provided by the National Ready Mixture Concrete Association (NRMCA) [131]. The calibration exercise utilized Case 1, a cubic meter of concrete, and the baseline scenario (Scenario 0). For calibration purposes, the workability functionality was disabled so that the optimization would only find solutions based on strength, cost, embodied energy, and embodied carbon. Cover depth was excluded due to the absence of a chloride hazard in Case 1. Additionally, functionality for replacement of coarse aggregate with recycled aggregates was disabled, as was accounting for sequestered carbon.

The results of the Case 1 Scenario 0 model and data comparison are shown in **Table 3-17** through **Table 3-19**. All outputs from the model were converted from

cubic meters to cubic yards for comparison. Statistical quantities where calculated from the set of Pareto-optimal solutions. The comparison with industry data is conducted to show that modeled optimal solutions lie within or near real-life data sets. The results indicate that the model outputs are comparable to industry data. All optimal mixture designs are of the same order of magnitude for each impact as those mixtures considered from industry. This calibration illustrates the ability of the model to report realistic mixture impacts.

Table 3-17. Model cost comparison with industry data

	Cost		
	Data	Model	
	$/yd^3	$/yd^3	% Difference
Lower Quartile	57.42	61.1	6.41%
Average	57.23	59.61	4.16%
Upper Quartile	56.39	57.27	1.56%

Table 3-18. Model embodied energy comparison with industry data

	EE		
	Data	Model	
	MJ/yd^3	MJ/yd^3	% Difference
Minimum	1304	1281	1.76%
Average	2211	1726	21.94%
Maximum	3518	2259	35.79%

Table 3-19. Model embodied carbon comparison with industry data

	EC		
	Data	Model	
	$kgCO_2/yd^3$	$kgCO_2/yd^3$	% Difference
Minimum	143	212	48.25%
Average	277	234	15.52%
Maximum	477	298	37.53%

While conducting the model calibration, interesting behavior of the model was discovered. When the model was first being developed, estimates of valid coefficients for the life cycle objectives, namely cost and embodied energy, were used as presented in **Table 3-20**. To calibrate the model, adjustments to many of the coefficients were made to match data found in literature (**Table 3-6**). As coefficients were individually altered, the model was run as an attempt at a pseudo-sensitivity analysis, which led to three initial insights into the model's behavior. The three alterations were to (1) reduce the distance value of SCMs by a factor of 10 from 500 to 50, (2) reduce the SCM replacement allowed from 30% to 0% and (3) lower the transportation cost coefficient by a factor of 50 to 0.0001 in order to match industry data. Each alteration was conducted independently of the others with the resulting solution field presented in **Figure 3-3**, **Figure 3-4** and **Figure 3-5** respectively, each utilize epsilons presented in **Section 3.3.2**. For reference, a plot of the modeling output using estimated coefficients is presented in **Figure 3-2**. Points on these plots correspond to individual mixture designs with shape corresponding to SCM type. More precisely triangles represent fly ash mixtures, circles silica fume, squares metakaolin and octagons slag.

Table 3-20. Estimated values of cost and energy impacts used during model development

Constituent	Cost ($/kg)	EEC (MJ/kg)	Distance (miles)
Coarse Agg	0.012	0.1	50
Fine Agg	0.02	0.08	75
Cement	0.0985	5.9	20
Water	0.005	0.01	20
Slag	0.017	1.6	500
Fly Ash	0.03	0.1	500
Silica Fume	0.18	0.04	500
Metakaolin	0.045	2.08	500
	($/kg-mi)	(MJ/kg-mi)	
Transportation	0.005	0.02	N/A

2D figure plots are utilized in this section and present solution mixes using four objective variables. The cost of the mix and embodied energy (EE) are shown on the x and y axes respectively, while strength is shown as a color corresponding to the adjoining color bar and SCM type is conveyed through the shape of the solution point, with triangles corresponding to fly ash mixes, circles silica fume, squares metakaolin and stars slag. The unaltered solution set, **Figure 3-2,** is shown to contain a solution of every SCM and consists of two separate solution fields. This can be seen through the grouping and type of solutions on the plot. The first field is the large central region containing solutions that include every SCM type and the other is the smaller conjoined region to the right of the central region consisting entirely of silica fume solutions.

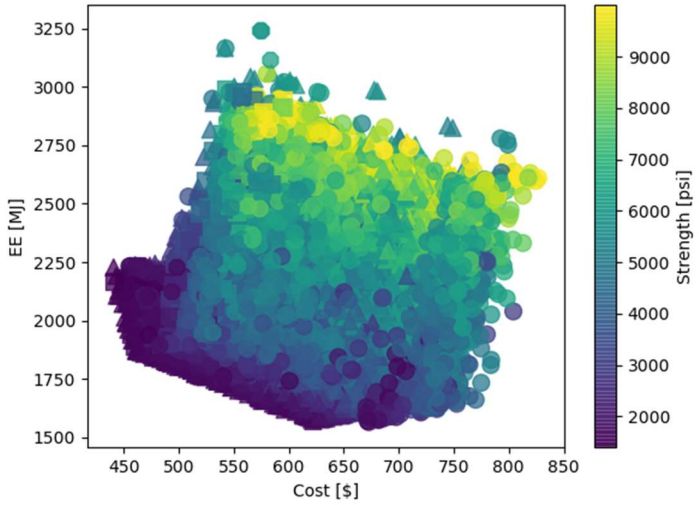

Figure 3-2. Solution field from Case 1 model using estimated coefficients during model development

As anticipated every alteration led to a reduced cost of the mixture to varying degrees compared to **Figure 3-2** with (1) and (2) having a <20% reduction while (3) led to a >80% reduction. The reduction in SCM travel distance (**Figure 3-3)** led to a 26% reduction in solution field area and also cut the SCMs of metakaolin and slag from the solution field. The resulting shape of the solution field appears to exclude all solutions corresponding to high embodied energy and low cost and a spatial distinction between fly ash and silica fume solution sets is created. Similarly, limiting SCM replacement to ~0% replacement (**Figure 3-4)** led to 30% reduction in solution field area with a similar shape formed to that of (1). Any comparison on the SCM types found in this solution set is not possible, the symbols indicating a particular SCM type for each solution is just a fragment of the formulation of the model that insists the MOEA choose an SCM type regardless of replacement choice, thus as expected all SCMs appear in the solution set.

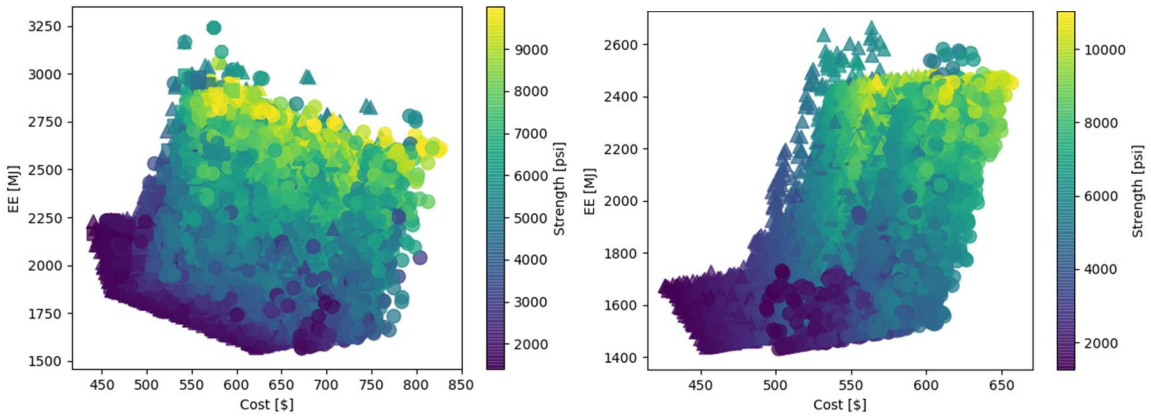

Figure 3-3. Solution field comparison for Case 1 model with estimated values (left) and reduced SCM distance (right)

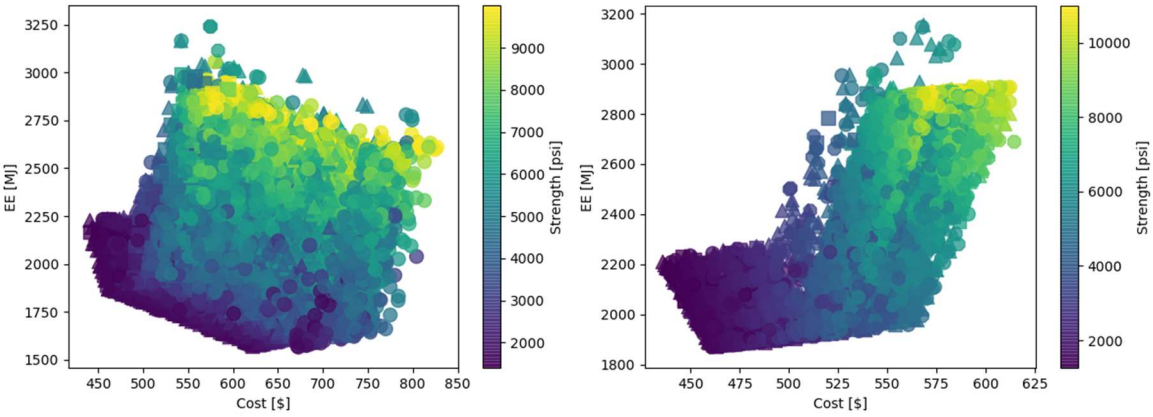

Figure 3-4. Solution field comparison for Case 1 model with estimated values (left) and zero SCM replacement (right)

With the greatest relative coefficient reduction, by a factor of 50, the transportation cost reduction led not only to the largest reduction in number of solutions (>50%) but also discontinuity in cost between discrete solution groups by SCM type as seen in **Figure 3-5**. Additionally, the SCMs of metakaolin and slag that were found in **Figure 3-2** were no longer present. These changes in the

solution space, while interesting, are primarily discussed to illustrate how simple input parameter changes can impact the solution field.

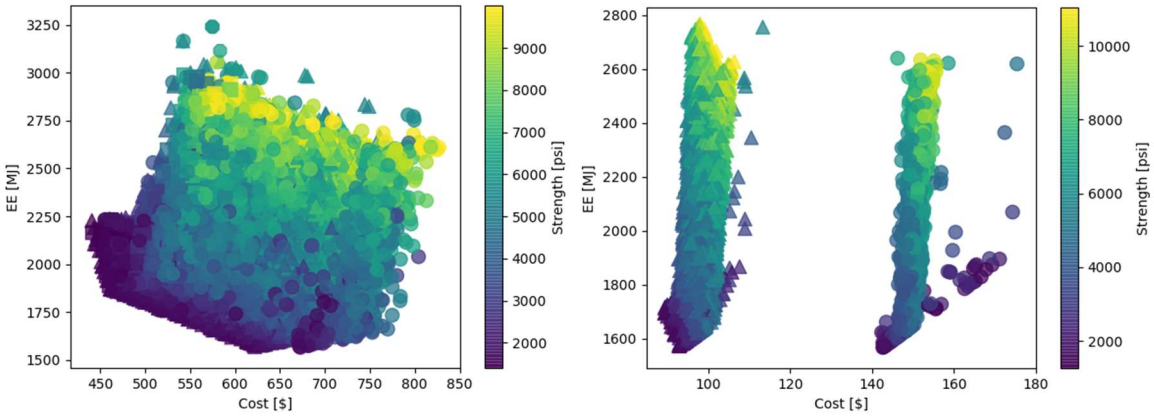

Figure 3-5. Solution field for Case 1 model with estimated inputs (right) and updated, industry, cost coefficient (left)

3.6.2. Case 1: Cubic Meter of Concrete

3D plots and parallel axis plots are used throughout this section to better convey the tradeoffs for each case. The 3D plots report cost, embodied energy, embodied carbon, and strength, while the parallel axis plots present either all six of the objective functions or the 13 objective functions and decision variables. Parallel axis plots of the objective functions are oriented such that the lower end of the axis is preferable. Due to the lowest values being considered preferable the plots display negative values for strength and slump such that the preferred values of each (the largest) are at the bottom of the axis. Each scenario is covered individually in numerical order per **Table 3-11** and compared to a baseline (**3.6.2.1**) and other cases, when applicable.

3.6.2.1. Case 1: Scenario 0 Baseline

As discussed in **Section 3.5**, a Case 1 Scenario 0 was created as a baseline solution set and is reported in **Figure 3-6**. Points correspond to individual mixture designs with shape corresponding to SCM type. More precisely, triangles and circles represent mixtures with fly ash and silica fume, respectively. A parallel axis plot is provided in conjunction with the 3D Cartesian plot to clearly communicate those objectives not seen in the 3D plot, cover and slump. For all subsequent parallel axis plots the blue solution lines represent the baseline, Scenario 0, solution set.

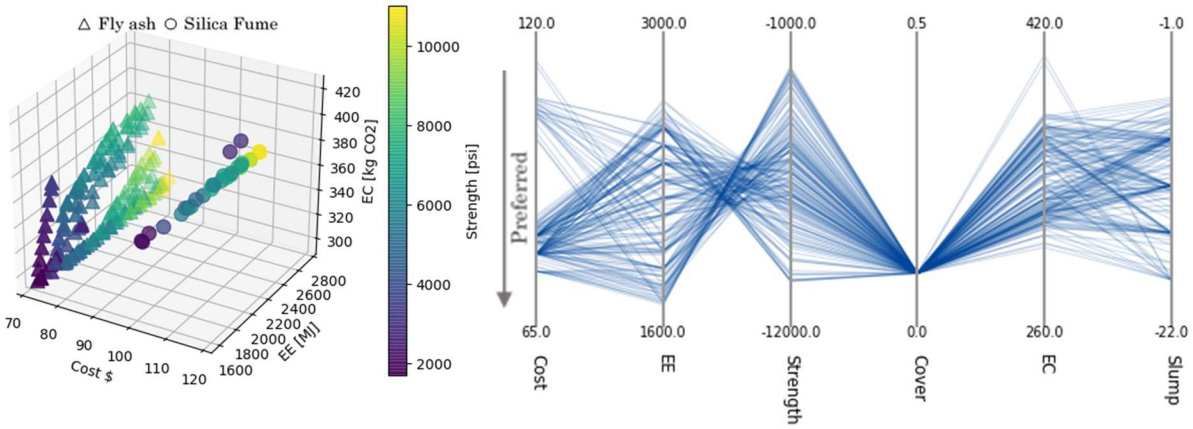

Figure 3-6. Case 1: Scenario 0 solution field and parallel axis plot.

The results show a similar grouping of solutions by SCM type as seen in Figure 3-5. As expected, cover depth is fixed on the parallel axis plot of **Figure 3-6**, since cover depth was held constant at 0.06 meters. Strength exhibits highest variation in possible values. The two isolated solution groups are distinguished by SCM type with only fly ash and silica fume appearing as optimal choices. As indicated by color, both solution groups indicate that, in general, an increase in strength led to an increase in embodied energy and carbon. This relationship is also indicated in the parallel axis plot. Crossed lines in the parallel axis plots indicate

proportionality between two objectives if one objective is positive and the other negative. This result is expected any time the variables mutually increase in magnitude, due to the counter orientation of the embodied energy and strength axes. In this particular example, average fly ash solutions were found to be 30% less costly than the silica fume solutions and experienced an average lower value of embodied carbon but greater variance as seen in the 3D plot of **Figure 3-6**. Workability appears to be optimal around a few values as shown in the parallel axis plot. This behavior is likely the result of the algorithm finding local extrema in the workability prediction function that allow numerous combinations of mixture proportions to result in similar workability. The dominance of only a few workability values is further illustrated in **Figure 3-7**.

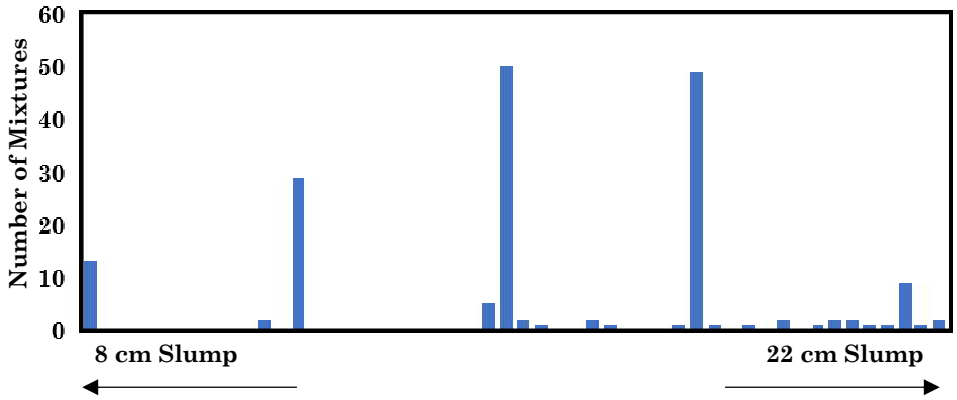

Figure 3-7. Histogram showing the significant grouping of workability values for Case 1 Scenario 0

3.6.2.2. Case 1: Scenario 1

The solution field from Case 0, Scenario 1, which exemplifies a possible increase in the price of fly ash to equal that of silica fume, is shown in **Figure 3-8**.

A parallel axis plot of the objectives is also provided in **Figure 3-8**. In such plots, each line represents an individual mixture design and each axis an objective variable. The red solution lines are solutions from this Scenario while the blue are the baseline, Scenario 0. The solutions are plotted over one another to indicate trade-offs and further communicate contrast between Scenarios.

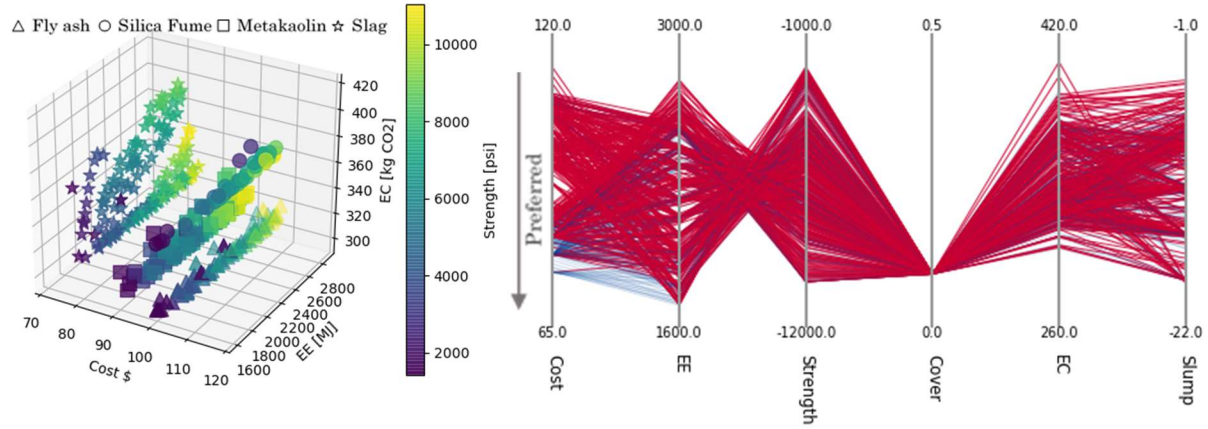

Figure 3-8. Case 1: Scenario 1 solution field and parallel axis plot.

For this particular case study example, results illustrate that, with an increased cost of fly ash, the field of possible solutions increases. This finding is demonstrated by the presence of all four SCMs in the solution set. Similar to the baseline case, solutions are distinctly grouped by SCM type in lines of constant cost as seen in the 3D plot of **Figure 3-8**. Additionally, slump appears to converge to four values, which is again attributable to consistent solution mix design quantities for cement, water, sand, and coarse aggregate. Changes in objective ranges and diversity are minimal, as only fly ash solutions experience increased cost. Similar grouping of slag solutions (stars) to the fly ash solutions seen previously in the baseline is attributed to similar coefficients for cost (**Table 3-6**) of the two

constituents. Most importantly, these results indicate that the model can account for future economic and production changes of an SCM.

3.6.2.3. Case 1: Scenario 2

The solution field resulting from an increase in embodied energy and carbon impacts by a factor of 20 for fly ash is shown **Figure 3-9**. Increasing the impact of fly ash by 20 times was done to allocate impacts from the burning of coal to fly ash such that the impact coefficients would be within the range of the other SCM impacts.

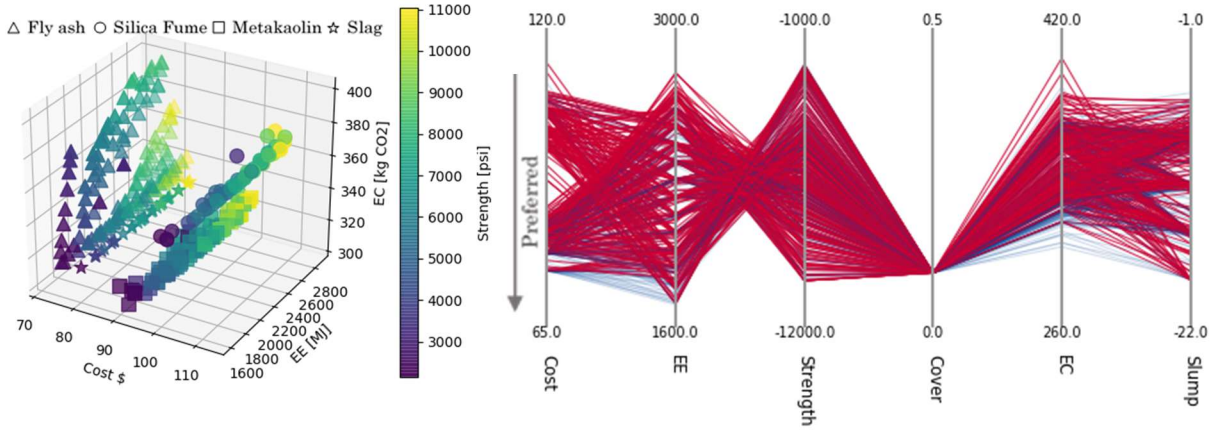

Figure 3-9. Case 1: Scenario 2 solution field and parallel axis plot.

Similar to the effects of increasing fly ash cost, the results show that potential mixture solution diversity is increased. Again, solutions are spatially separated by SCM type with metakaolin (squares) and slag (stars) appearing in the solution set. Fly ash solutions occupy the same region and distribution as in the base case while slag solutions, previously seen in Case 1 to be non-dominated at low cost, are reduced to a minimal region at the base of the fly ash solutions. As in Case 1, this finding is attributable to the relatively similar cost of fly ash and slag which permits slag solutions to exist either when fly ash is more expensive or has greater

embodied impacts. As expected, metakaolin and silica fume solutions are unchanged from Scenario 1, as are the range and diversity of objectives. These findings further reinforce those of Scenario 1, where slump was also grouped at a few distinct values, indicating that only a few dominant mixture solutions exist. Even with a twenty-fold increase in embodied impacts fly ash did not lose preference in this solution set. This result indicates that the precise impact accounting, used to formulate LCI values, for fly ash is less impactful than more precise accounting for the other SCMs. Thus, variation in the data set and techniques used to define fly ash impacts is relatively unimportant for this specific model but could prove more impactful in other scenarios or case's.

3.6.2.4. Case 1: Scenario 3

The impact of closer proximity to recycled aggregates on the solution field is shown in **Figure 3-10**. A parallel axis plot of both decisions and objectives is provided with each line representing an individual mixture design and each axis an objective or decision variable. This was done to clearly illustrate the range of recycled aggregates chosen by the model. Although the cost and environmental impacts of recycled aggregates were designated to be lower than normal aggregates (**Table 3-6**), their distance was specified as greater as they are less prevalent and processed differently. By decreasing the distance to be a fifth that of normal aggregate, recycled aggregate use in mixtures was anticipated to increase.

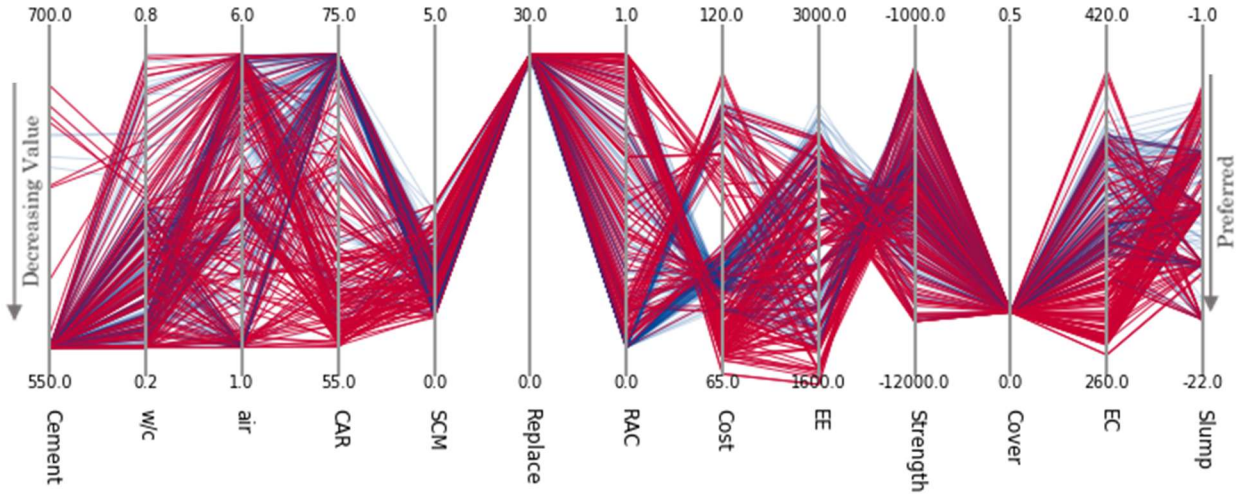

Figure 3-10. Case 1: Scenario 3 objective and decision parallel axis plot.

Results indicate that the reduced distance of travel for recycled aggregates by 90% results in the full available range of recycled aggregate replacement in potential mix design solutions. Additionally, coarse aggregate content is shown to often be chosen as the largest possible value. This aligns with the algorithm choosing to maximize recycled aggregate content, since recycled aggregates are only considered to replace coarse aggregates in this model. Although no four dominant mixture designs appear to exist on the decisions (right) side of **Figure 3-10**, an even more pronounced grouping of slump values is reported. This result further indicates the possibility that the MOEA algorithm may have found local extrema in the workability prediction relationship that allows diverse mixture designs to result in the same predicted slump. **Figure 3-11 - 12** are provided to clearly indicate the diversity of mixture proportions, namely water and coarse aggregate in the above solutions although there is grouping of workability values similar to **3.6.2.1**. These two mixture constituents were not chosen to be exhaustive, but rather, were chosen because they have the most beneficial effect on workability of all constituents. Thus, through the observed variability in these two constituents, it can be assumed that a

highly diverse set of mixtures produced the limited number of workability values seen in **Figure 3-10**.

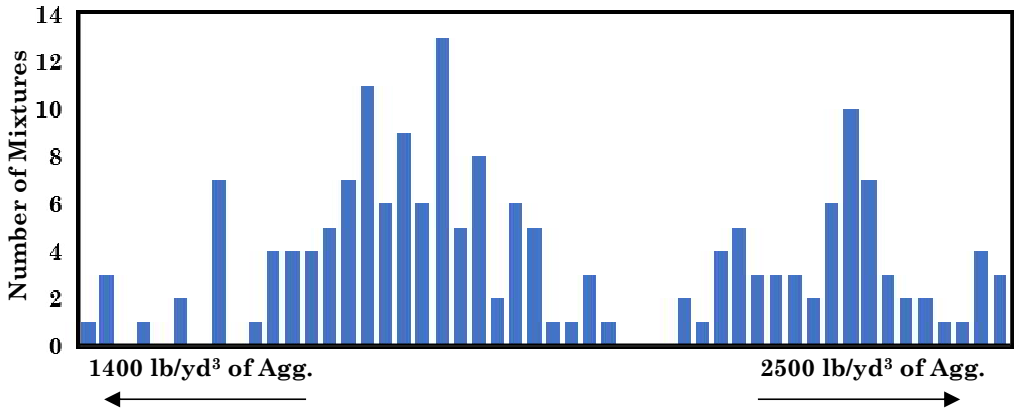

Figure 3-11. Histogram of Coarse Aggregate Content of Solution Mixtures in Case 1 Scenario 3

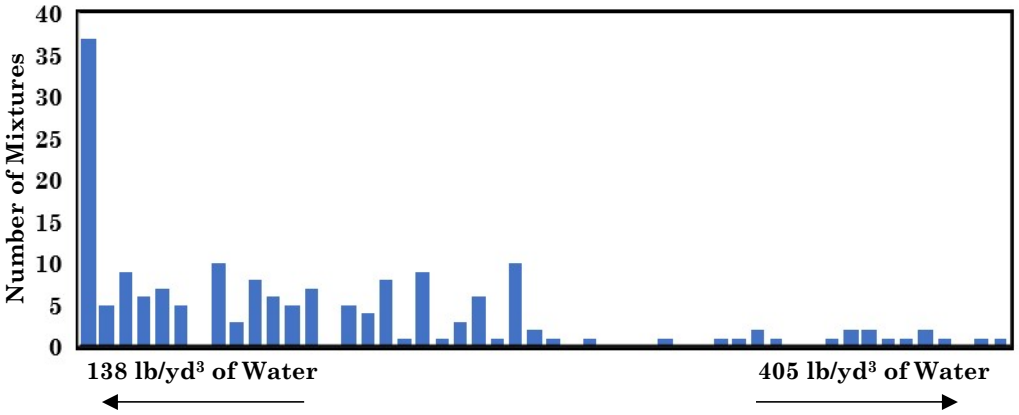

Figure 3-12. Histogram of Water Content of Solution Mixtures in Case 1 Scenario 3

Compared to the baseline solution set, the mixtures containing recycled aggregates have lower cost by >7% with solutions being predominately below $75 in cost. Embodied energy and carbon impacts, although similar in maximum solution values, have reduced minimums by <10%. Although 100% replacement by recycled

aggregate is not specified for every mixture, very high recycle content mixes constitute a plurality with >30% of the mixtures containing 99% recycled aggregate or more. These results show that, without consideration for durability, use of recycled aggregates can be considered in the optimal solution set in this case study example, given they are more accessible than their virgin counterparts. Additionally, results indicate the importance of transportation costs and impacts when accounting for a low-impact constituent, such as recycled aggregates.

3.6.2.5. Case 1: Scenario 4

The solution field resulting from modifications to transportation distances for each SCM (see **Table 3-14**) is shown in **Figure 3-13**. Additionally, a parallel axis plot of the objectives is provided with each line representing an individual mixture design and each axis an objective variable.

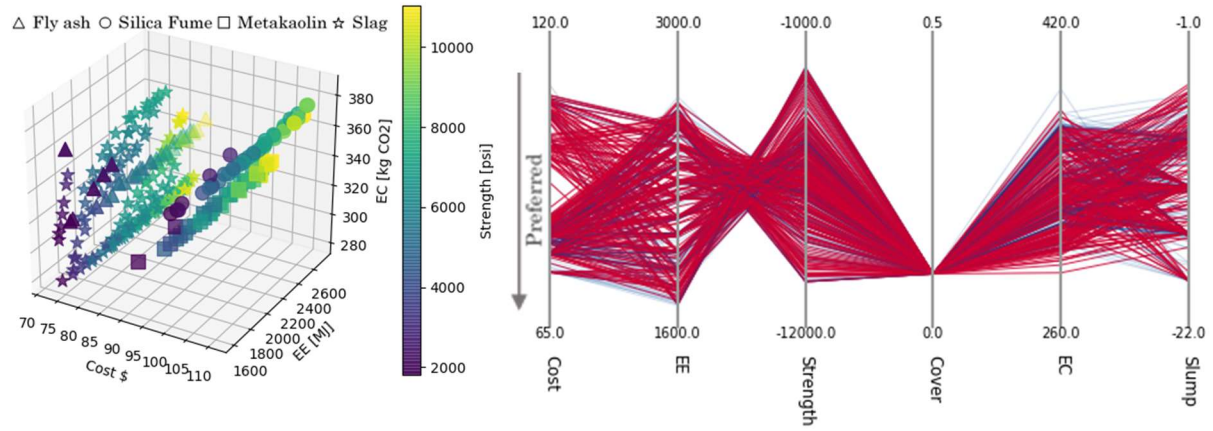

Figure 3-13. Case 1: Scenario 4 solution field and parallel axis plot.

Distances used for each SCM indicate, as seen previously, that fly ash and silica fume are preferred to metakaolin and slag in Case 1. The preference is indicated, by the 200% increase in fly ash distance and 50% increase in silica fume distance needed to bring the other SCMs into the solution space. The grouping of

slump values at nearly the same solution points to the previous scenarios is clear. Results indicate that solutions are grouped by SCM, although, as first noted in Scenario 2, slag and fly ash solutions are highly linked in this example and, thus, closely grouped in **Figure 3-13**. Although results show solution groupings similar to those in Scenario 1 and Scenario 2, the magnitudes and ranges of objectives, namely embodied energy and carbon are reduced, with each having reduced maximum solution values. Again, these results illustrate the significant impact of transportation on life cycle impacts.

3.6.2.6. Case 1: Scenario 5

The solution field resulting from a train as the means of transportation rather than a truck is displayed in **Figure 3-14**. Additionally, a parallel axis plot of the objectives is provided with each line representing an individual mixture design and each axis an objective function. A train was chosen as they are commonly used to transport bulk materials around the United States and to batch plants.

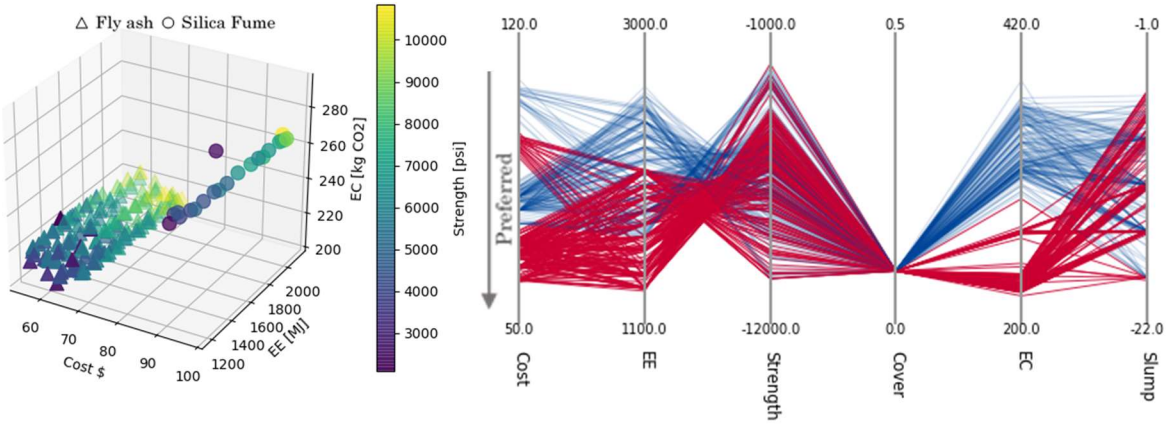

Figure 3-14. Case 1: Scenario 5 solution set and parallel axis plot.

As expected, results show that transportation coefficients have large impacts on the solution set output as first seen in **Figure 3-5**. The average value of the

range of objective functions decreased for cost, energy, and carbon by a 15%, 12% and 29% respectively with embodied carbon also experiencing a significant decrease in solution diversity compared with the baseline. While the overall reductions in magnitude of solutions are representative of the decrease in transportation cost alone, solution diversity reduction is considered to result from the dominant role that transportation plays in shaping the solution field of low-impact constituents similar to what the results in Figure 3-10 indicated. Additionally, embodied carbon impacts were heavily stratified by SCM type, with all silica fume solutions contributing on average 40 kg-CO_2e (or 20%) more than fly ash solutions. This result is attributable to the increased contribution of constituent coefficients to total impacts as transportation impacts decrease. As shown in **Figure 3-15**, this scenario produced a high usage of recycled aggregate similar to section **3.6.2.4** with 20% of solutions having replacements over 90%. These results further reinforce the conclusion that low impact constituents are most impacted by transportation impacts and distances.

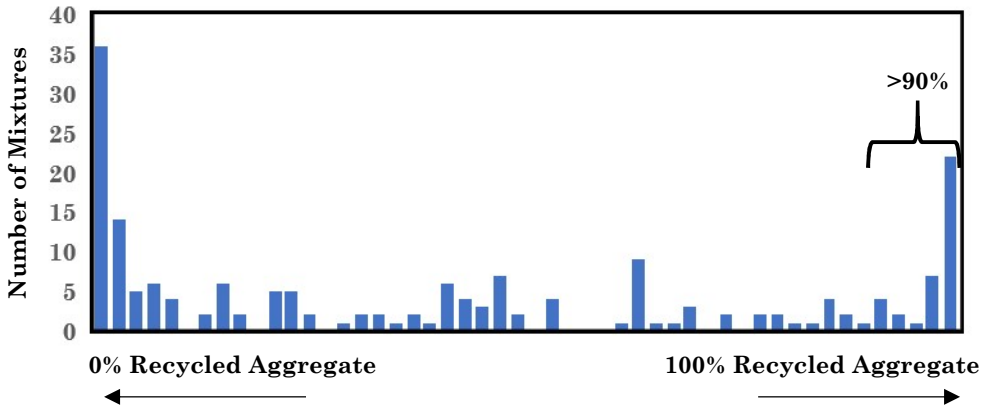

Figure 3-15. Histogram of Recycled Aggregate Content in Solutions for Case 1 Scenario 5

3.6.2.7. Case 1: Scenario 6

The solution set resulting from an increase in the lower bound of the w/c ratio is presented in **Figure 3-16**. Additionally, a parallel axis plot of the objectives is provided, with each line representing an individual mixture design and each axis an objective function.

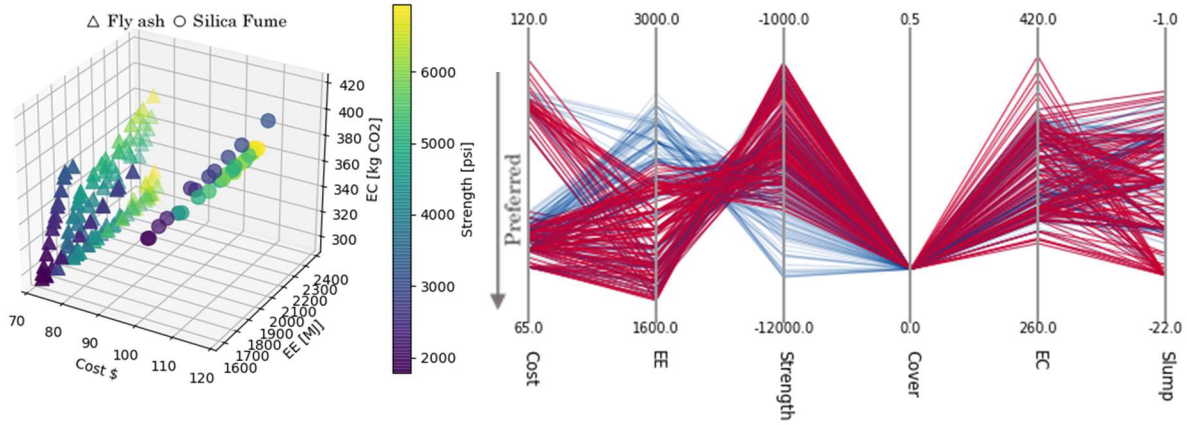

Figure 3-16. Case 1: Scenario 6 solution set and parallel axis plot.

Results indicate, as anticipated, that increasing the minimum allowable w/c ratio for the model reduces the resulting compressive strengths of mixture design solutions compared to the baseline model. Surprisingly, neither cost (< 2%) nor embodied carbon (slightly < 7%) are significantly affected. Conversely, maximum embodied energy values for each mixture are reduced by 400 MJ (14%), which parallels the 36% reduction in maximum compressive strength. Solution field ranges for the remaining objectives are not significantly altered from the baseline. This finding is attributable to the lack of accounting for durability, which is heavily influenced by w/c ratio. As with previous results, this case produces similar groupings of slumps at four dominant values and also exhibits an increased use of recycled aggregates, although not as pronounced as in Case 1, Scenario 3. The

presence of recycled aggregates implies that, without accounting for durability, low w/c ratio mixes could utilize recycled aggregates to effectively reduce their cost and embodied impacts.

3.6.3. Case 2: Concrete Tilt-Up Wall

Similar to **Section 3.6.2**, the results for the concrete tilt-up wall case study are documented with a combination of 3D and parallel axes plots. Both simplified and numerical 1D diffusion models are employed throughout this section, and their comparison is included when noted. In general, the numerical 1D diffusion model is the default case and used for the following cases, which are presented in order per **Table 3-11**. A baseline case (Scenario 0) is first discussed and utilized to compare with other case results. Due to the limited visual complexity of solution sets on the 3D plots, results for Scenario 1 and Scenario 2 are presented with parallel axis plots containing both objective and decision variable axes, which necessitated the use of these more complete parallel axis plots throughout this section. Each case is covered individually in numerical order per **Table 3-11** and compared to the baseline and other cases when applicable.

3.6.3.1. Case 2: Scenario 0 Baseline

Formulation of the base case solution was performed for both the simplified and numerical 1D diffusion model and is shown with 3D plots in **Figure 3-17**. Modeling parameters specified in **Table 3-15** were used in the corresponding models. Additionally, a parallel axis pot of the objectives and decisions is provided with each line representing an individual mixture design and each axis an objective or decision variable. Throughout Case 2 investigations, the first seven axes are labeled for decisions (e.g., cement content, w/c ratio, air content) and the other six axes are labeled with objectives (e.g., cost, embodied energy, strength). These

parallel axis plots are plotted for both the simplified model and numerical 1D model (**Figure 3-18**).

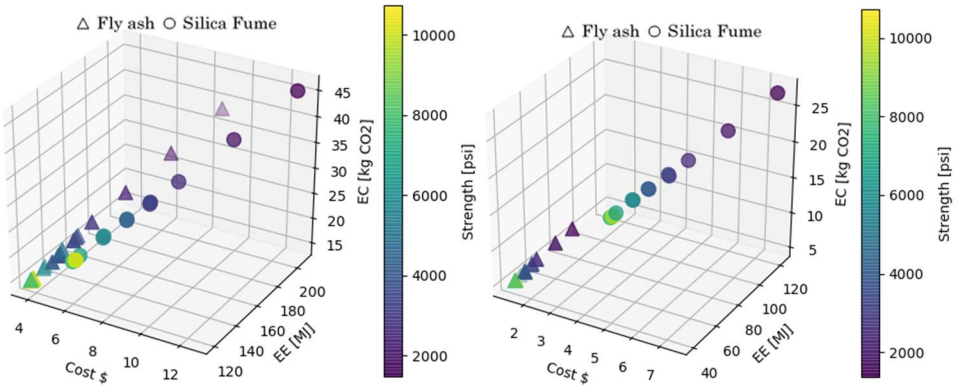

Figure 3-17. Case 2: Scenario 0 solution field using a simplified (left) or numerical (right) chloride diffusion model.

The results in **Figure 3-17** indicate that, regardless of chloride transport model, there is a linear relationship between cost, embodied energy, and embodied carbon. Contrary to the findings of the Case 1 (cubic meter) baseline (**3.6.2.1**), strength, cost, and embodied carbon appear to be inversely proportional to one another, such that an increase in strength will result in a decrease in cost, embodied energy, and embodied carbon. The linear relationship for Case 2 is true regardless of the diffusion model. This result is expected, as both diffusion models predict lower diffusion coefficients and, thus, lower cover depths for lower w/c ratios, which, in turn are responsible for higher strengths. With the only variable being cover depth required to resist corrosion for 100 years, the lower the cover, the lower the cost, embodied energy, and embodied carbon of the mixture used to create it. Results of the numerical 1D model indicate that solutions utilizing fly ash occupy a different range of cost, embodied energy, and embodied carbon than those of silica

fume, while the simplified model indicates that both occupy nearly the same range. This finding illustrates the greater impact that fly ash imparts on the diffusion coefficient with time, something not accounted for in the simplified model. Therefore, fly ash-containing can improve the performance of mixtures beyond the range of silica fume-containing mixtures. Additionally, cost, embodied energy, and embodied carbon are noted to be approximately 50% less for each metric compared to the simplified diffusion model, which is attributable to the known conservative cover calculation produced by the simplified approach.

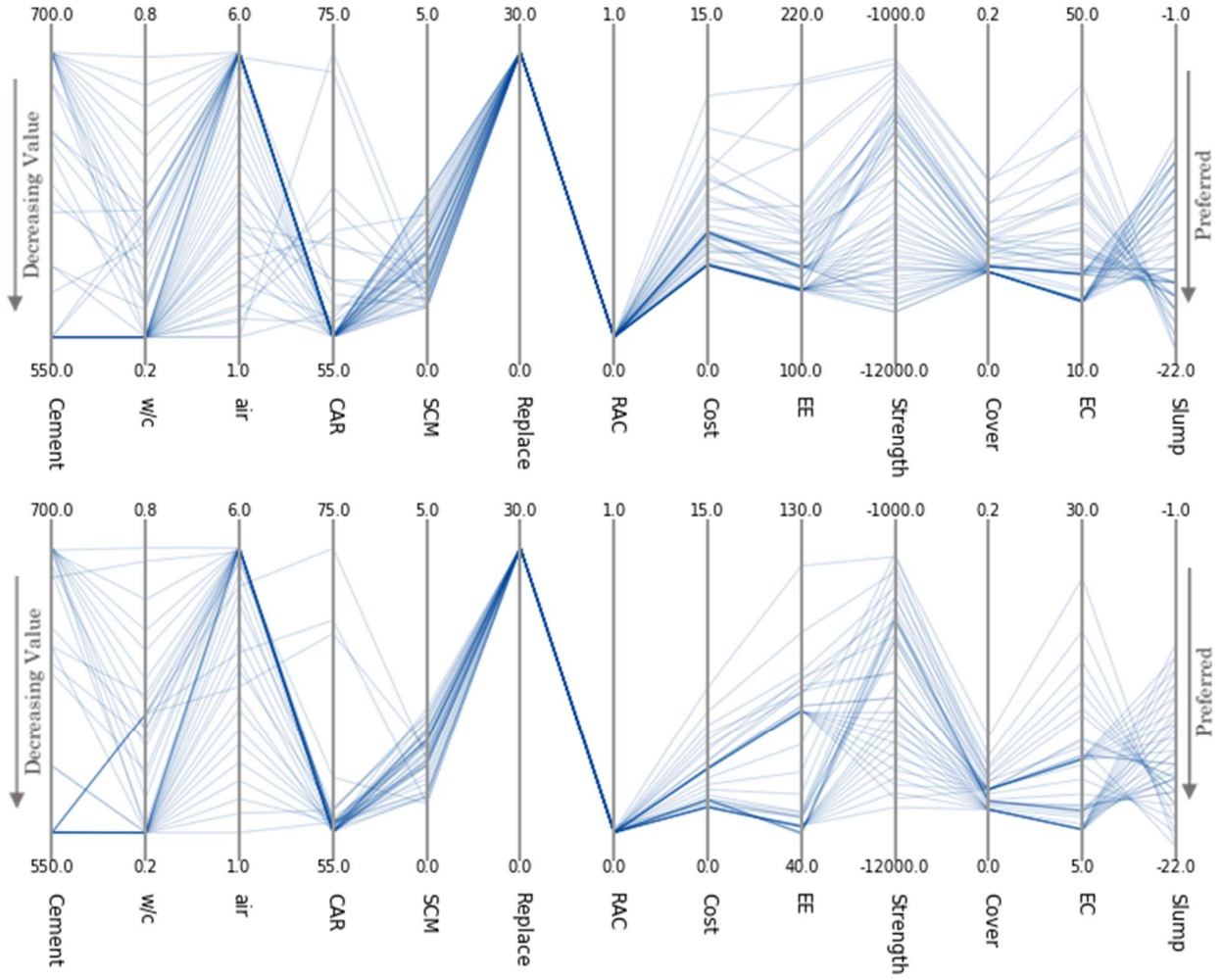

Figure 3-18. Case 2 scenario 0 objective and decision parallel axis plots with simplified diffusion (top) and numerical 1D diffusion (bottom)

The parallel axis plot results confirm many of the conclusions drawn from **Figure 3-17**. As expected, the cover for the numerical service life model is, on average, lower by 40% than the simplified model. Additionally, results show the exclusive choice of 30% SCM replacement and lack of recycled aggregate replacement, as seen previously with the cubic meter functional unit. This investigation illustrates the undesirability of recycled aggregates for their negative impact on durability, as found in Case 1. However, additional investigation is

needed to illustrate the distances, costs, and availability thresholds that would render recycled aggregates preferable in chloride-laden environments.

The results in **Figure 3-18** also indicate that slump is inversely proportional to the embodied energy, cover, embodied carbon, and cost, while again being directly proportional to strength. Similarly, from Case 1 to Case 2, the embodied energy and, by extension, cost and embodied carbon, are inversely proportional to strength, as discussed previously. Unlike Case 1 (cubic meter), the slump solutions no longer group at values and are instead spread fairly uniformly across the specified range. This result is attributable to the reduction in the choice of epsilon for the objective function of slump between Case 1 and Case 2. The grouping of solutions along the cover axis is indicative of the diffusion coefficient modifiers for fly ash and silica fume driving the cover calculation. Furthermore, the increased temporal degradation of diffusion coefficient due to fly ash inclusion considered in this model results in reduced cover over the silica fume solutions, as speculated in previous work by Srubar [17] and in **Chapter 2**. Unexpectedly, the results indicate that all w/c ratios can be used in optimal solutions. Although the majority of mixtures (61%) choose the lowest w/c ratio, those that do not end up with some of the highest slumps > 19 cm of the mixes. This result confirms the value of including workability relationships in the model, as this behavior would likely not be exhibited without such models. Since air is modeled to only impact the strength of the mixture, along with cement and w/c ratio, it is not surprising that the results contain air contents that vary throughout the specified range.

3.6.3.2. Case 2: Scenario 1

The solution field resulting from an increase in fly ash price, such that it is as expensive as silica fume, is shown in **Figure 3-19**. The numerical 1D diffusion model was utilized to produce these results. A parallel axis pot of the objectives and

decisions is provided with each line representing an individual mixture design and each axis an objective or decision variable.

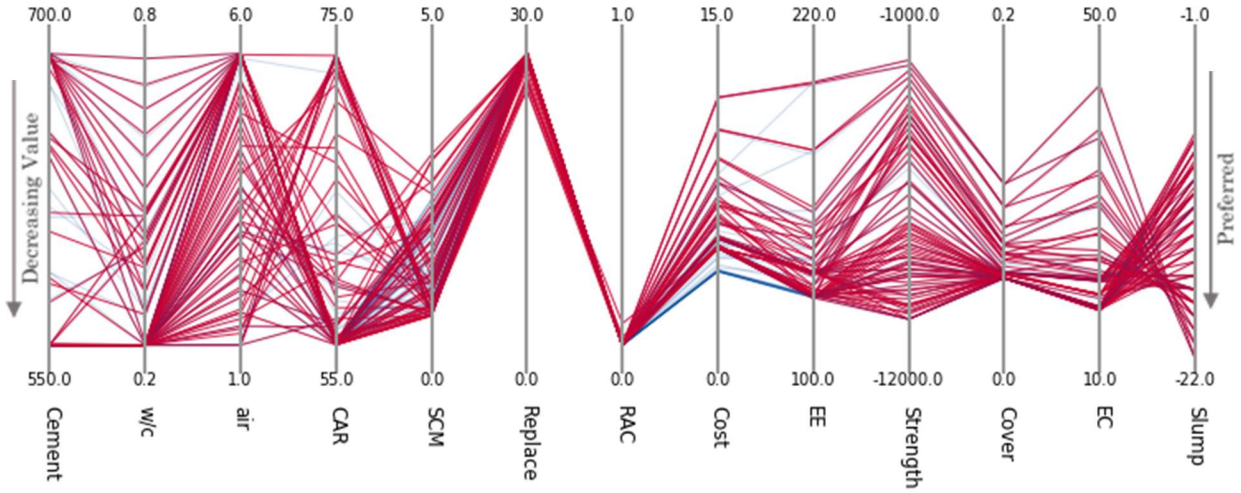

Figure 3-19. Case 2: Scenario 1 objective and decision parallel axis plot.

Results indicate continuation of many of the trends witnessed in the tilt-up wall baseline with zero aggregate replacement, inversely proportional cost, energy, and carbon impacts on strength, a uniformly distributed slump, and wide ranges of w/c ratios. As expected, the increase in fly ash price resulted in the inclusion of additional SCMs in the solution set, namely metakaolin. As in **Chapter 2** the absence of slag in the solution set is attributable to its minimal effects on the diffusion coefficient compared to the other SCMs in the relationships utilized to inform the model. Increased coarse aggregate ratios are also seen in the results but only for fly ash mixes. Most surprising of the results in **Figure 3-19** is the single solution that contains the maximum cement content, w/c ratio, air content, and coarse aggregate ratio, which are conflicting decisions for objectives, such as strength and slump and yet are considered an optimal solution. Results such as this

indicate the valuable capability of the model to elucidate tradeoffs that might not be discovered in experimental work.

3.6.3.3. Case 2: Scenario 2

The solution set resulting from an increase in fly ash environmental impacts to the levels shown in **Table 3-12** are presented in **Figure 3-20**. A 1D numerical model was utilized to produce these results. A parallel axis plot of the objectives and decisions is provided in which each line represents an individual mixture design and each axis corresponds to an objective or decision variable.

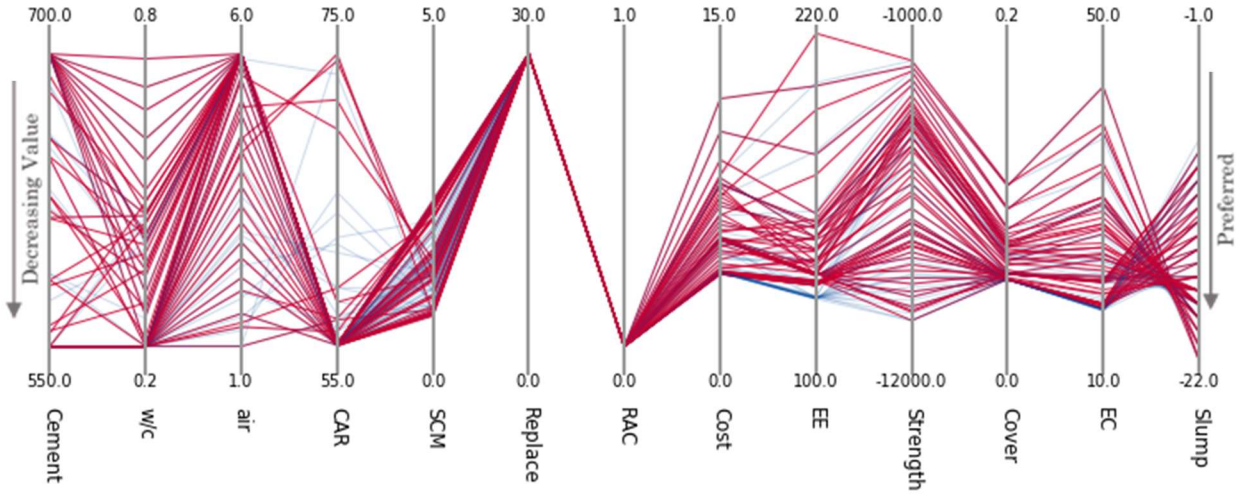

Figure 3-20. Case 2: Scenario 2 objective and decision parallel axis plot.

As expected the results are highly correlated to those of Case 1 with objective performance not significantly changed from the baseline and decisions including high diversity for cement content. Unlike in **Figure 3-19** the change in fly ash embodied impacts does not result in a significant increase in coarse aggregate replacement. The comparison of this result and that of Scenario 1 indicates that the use of a lower cost constituent, coarse aggregate, as filler is not beneficial within the increase in embodied impacts investigation (Scenario 2) as in the increase in cost

investigation (Scenario 1). While in previous investigations the changes to fly ash cost and embodied impacts resulted in increased inclusion of SCMs in the solution field, this behavior is not evident in this scenario, as neither metakaolin nor slag are present in the results. Instead, fly ash and silica fume are only utilized in the mixtures. Cover is primarily grouped at two values, as seen in Scenario 0.

3.6.3.4. Case 2: Scenario 3

Specifying the recycled aggregate content to be 100% forced the model to produce solutions containing recycled aggregates. Results were produced utilizing the numerical 1D diffusion model utilizing the same distance coefficient specified in **Table 3-12** and are presented in **Figure 3-21** and **Figure 3-22**. On the 3D plot points correspond to individual mixture designs with shape corresponding to SCM type. More precisely, triangles represent fly ash mixtures and circles, silica fume. Additionally, a parallel axis pot of the objectives and decisions is provided with each line representing an individual mixture design and each axis an objective or decision variable. The first seven axes are labeled for decisions (cement content, w/c ratio, air content etc.) and the concluding six axes are labeled with objectives (cost, embodied energy, strength, etc.). The red line indicates the Scenario 3 solutions while the blue line are the baseline (Scenario 0) solutions.

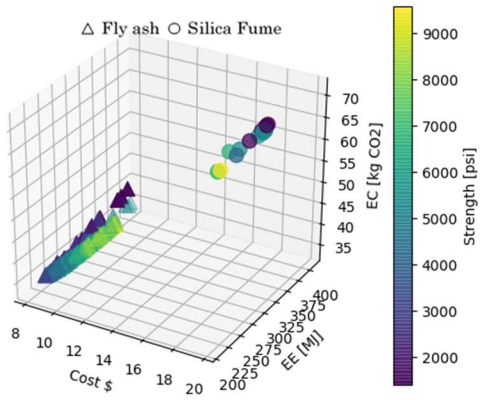

Figure 3-21. Case 2: Scenario 3 solution field.

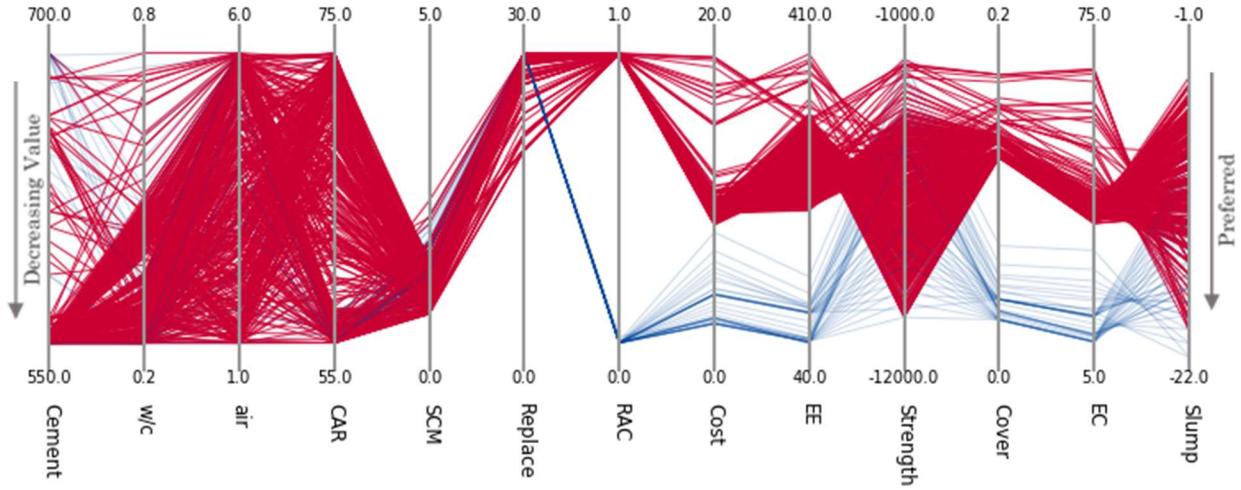

Figure 3-22. Case 2: Scenario 3 objective and decision parallel plot.

As expected the forcing of recycled aggregates at 100% replacement in all mixtures resulted in a much more extreme solution field than of the baseline. Cover was found to increase by more than 200% compared to **Figure 3-18** with maximum covers reported at over .15m or 5.9 inches. This finding agrees with results presented in **Chapter 2** that recycled aggregate replacement is a significant driver of chloride durability for concrete. Cost, embodied energy and carbon also increased by 200% over the baseline because of increased cover. The decrease in durability

that leads to an increased cover needed counteracts the savings that are assumed to come from the inclusion of recycled aggregates and is the justification for why recycled aggregates were not being utilized by the model previously. Reductions in strength are negligible compared to the baseline but could result from the higher density of solutions utilizing w/c ratios above 0.25. Air content and coarse aggregate content are both shown to be highly diverse with solutions occupying their full allowable range. Finally, slump values are noted to be lower on average by 7 cm with most solutions having slumps below 15cm or 5inches. The decrease in solution slump is contrary to previous solutions in **Figure 3-10** that have shown insignificant change in slump with recycled aggregate addition.

Results for the solution sets replacement percentage surprisingly indicate usage of SCM below the 30% replacement. The selection of lower than 30% replacement values in the optimal set could be a result of the model cutting SCM replacement to save costs in solutions were further replacement wouldn't significant change cover depth. These results show the potential for the model to find solutions even when a decision variable, such as recycled aggregate replacement, is forced to a designer specified value. Furthermore, this investigation illustrates that the tool is capable of illustrating trade-offs for scenarios that would otherwise be unlikely to be investigated in experimental research.

3.6.3.5. Case 2: Scenario 4

Scenario 4 was excluded from these results as no feasible distance (>0 miles) for slag could be chosen such that it would appear in the solution field. Additionally, the distances required for metakaolin to be used in solutions required fly ash to be over 10,000 miles from the mixing site, which is more than three times the width of the United States. Given these two findings, Case 4 is considered to not be attainable for the tilt-up wall functional unit modeled with the inputs presented in

Table 3-15. Additional investigation of less harsh environments with longer service life may result in slag becoming a solution option with only exposure Case 4 and 5 having the potential per the findings of **Chapter 2** in **Table 2-7**.

3.6.3.6. Case 2: Scenario 5

Reduction in transportation costs and impacts, through the use of a freight train rather than truck on the mixture design for a tilt-up wall, is reported in **Figure 3-23**. Red solutions are those utilizing train transportation while blue are those utilizing truck transport. Again, the 1D diffusion model was utilized to produce the results. A parallel axis plot of the objectives and decisions is provided with each line representing an individual mixture design and each axis a variable.

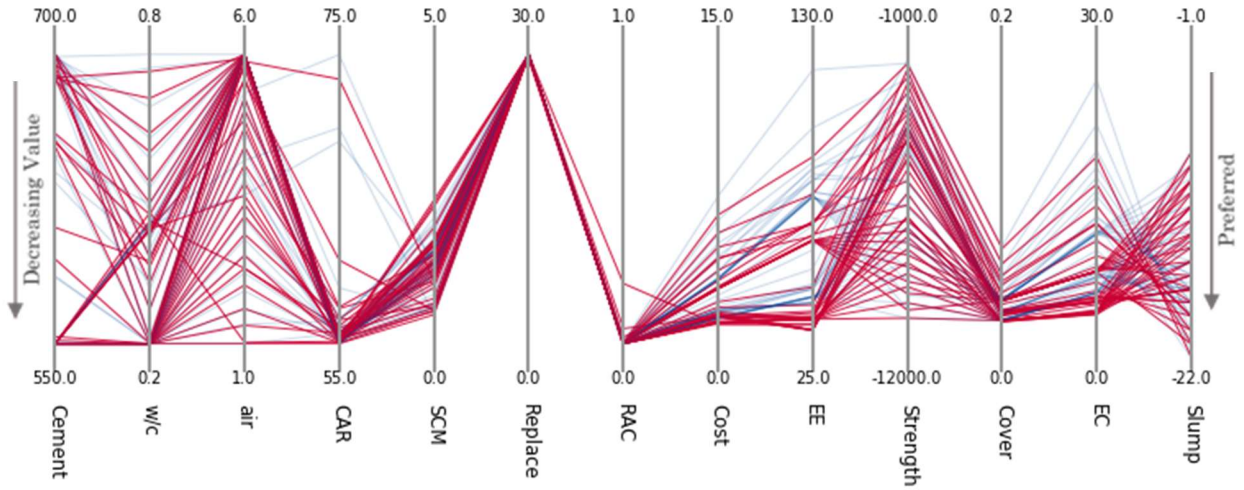

Figure 3-23. Case 2: Scenario 5 objective and decision parallel plot.

From the results, many of the same behaviors as previously discussed are present. For example, results indicate, a fully utilized SCM replacement amongst solutions, even distribution of slump in solutions, and full diversity in solution air content. As with the previous investigation of utilizing reduced transportation impacts (**Figure 3-14**), the results indicate a universal reduction in cost, embodied

energy, and embodied carbon by >20% each. Furthermore, greater variation between the fly ash solution group and silica fume solution group are seen in the results (**Figure 3-22**) as were seen previously through this case investigation. The presence of grouped solutions, again indicates the greater impact transportation has on fly ash as compared to silica fume in the model.

3.6.3.7. Case 2: Scenario 6

Results from increasing the lower bound of w/c ratio for a tilt-up wall are presented in Figure 3-24 and **Figure 3-25**. On the 3D plots, points correspond to individual mixture designs with shape corresponding to SCM type. More precisely, triangles represent fly ash mixtures and circles silica fume. Additionally, a parallel axis pot of the objectives and decisions is provided with each line representing an individual mixture design and each axis an objective or decision variable. The first seven axes are labeled for decisions (cement content, w/c ratio, air content etc.) and the concluding six axes are labeled with objectives (cost, embodied energy, strength, etc.).

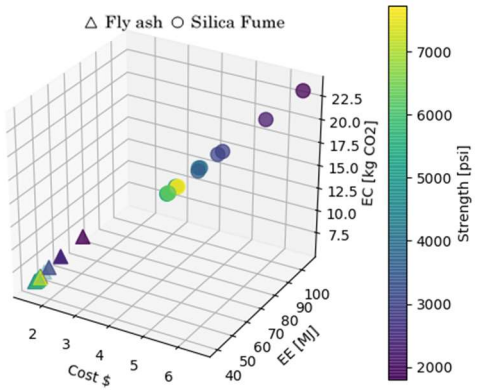

Figure 3-24. Case 2: Scenario 6 solution field.

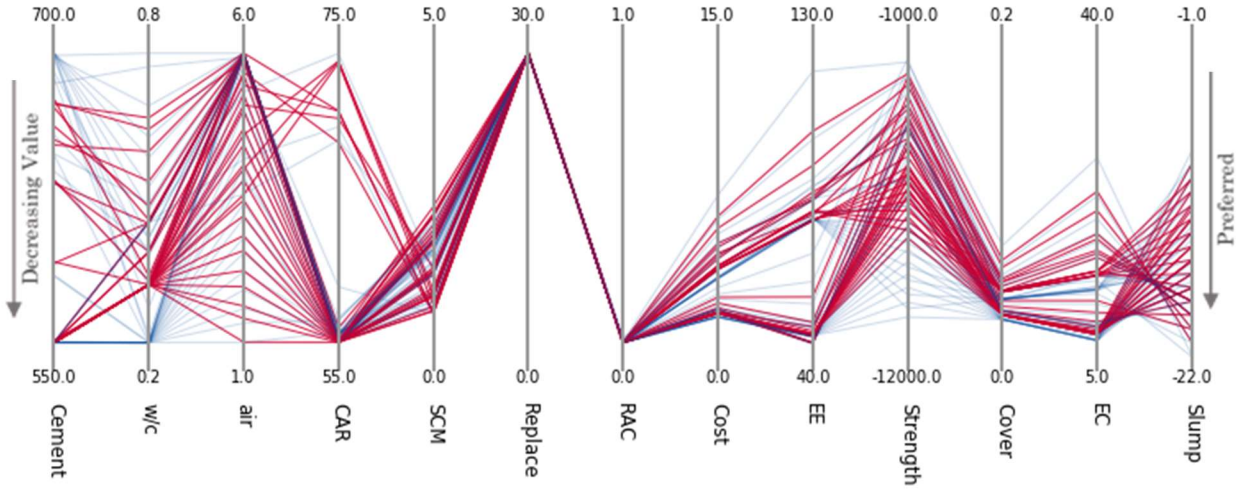

Figure 3-25. Case 2: Scenario 6 objective and decision parallel plot.

Results reiterate that increasing the w/c ratio lower bound produces a weaker and less varied solution set. Compared to the baseline, maximum strength is reduced by 22%, and the two solution groups corresponding to fly ash and silica fume are more separated. Reductions are also present in embodied carbon and energy on the order of 5% and 7%, respectively, for their minimums. No significant changes are noted for either cover or slump which suggests lower water-to-cement ratios are countered by other variables that can improve these properties allowing the tool to specify mixtures for lower strengths that have similar properties to lower w/c ratio mixtures. The reduction in w/c ratio is shown, through the results, to also reduce the cement content of the solution sets. but has no impact on the potential air contents.

3.6.4. Case 3: Axially Loaded Concrete Column

Results for Case 3 are presented for three scenarios: Scenario 0, Scenario 1C and Scenario 2C. This section utilizes both 3D plots of solution sets and combined objective and decision parallel axis plots that follow the same conventions applied for plots of Case 1 and Case 2.

3.6.4.1. Case 3: Scenario 0 Baseline

The baseline scenario for Case 3, the axially loaded concrete column, was created with the inputs in **Table 3-16**. The results are shown in **Figure 3-26** and **Figure 3-27**. On the 3D plot, points correspond to individual mixture designs with shape corresponding to SCM type. More precisely, triangles represent fly ash mixtures, and circles represent silica fume. Each line in the parallel axis plot of the objectives and decisions represent an individual mixture design, and each axis represents an objective or decision variable.

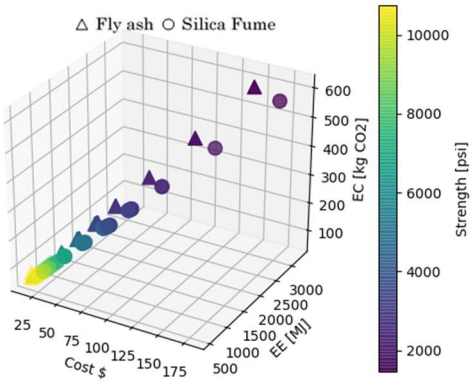

Figure 3-26. 3D plotted solutions for the baseline column

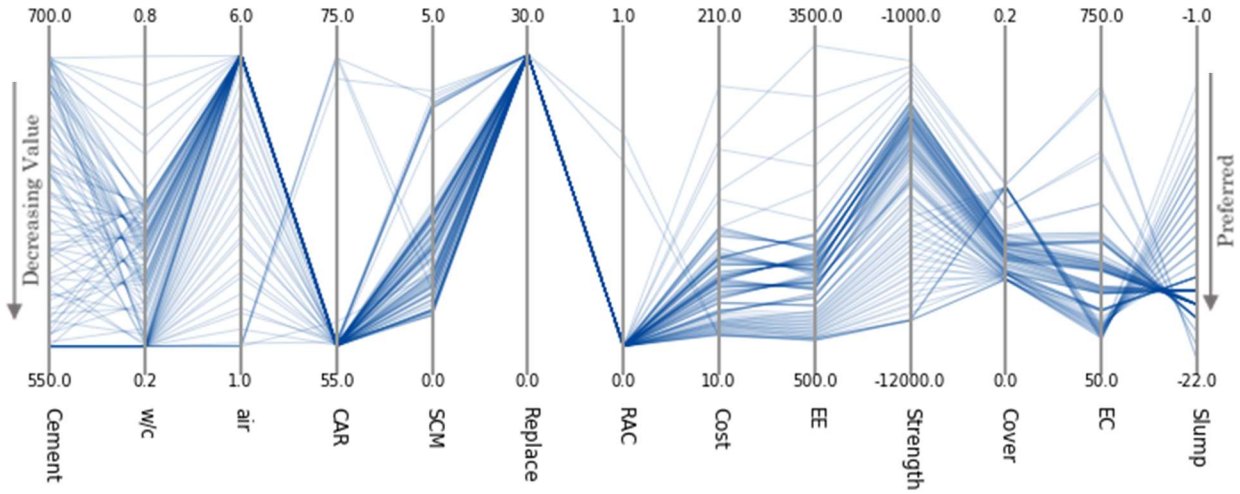

Figure 3-27. Parallel plot of solution set for the baseline column

Results indicate the presence of slag in the solution set that is running a numerical diffusion model and has standard inputs. This baseline case is the only one in this study to find slag-containing mixture design solutions. This finding illustrates the capability of the model, together with improved service life analysis, to produce more diverse solution sets. The diversity of cement, w/c, and air content in the solution set is high, which follows trends observed in previous results. Although most solutions favor low coarse aggregate content with no recycled aggregate, a few mixture solutions utilize high values of both, which led to the lowest cost mixtures, but the highest cover (>100% from the median cover of the solutions). These results indicate the possibility of some mixtures to contain recycled aggregates (even in moderate chloride environments) if larger covers are considered acceptable. Additionally, as seen in previous investigations, those solutions that contain the highest possible contents of cement, w/c ratio, and air have the highest cost, embodied energy, and embodied carbon and the lowest strengths, but produce the highest slump mixtures. The presence of these solutions helps indicate conflict between workability and the other objectives, and again shows that the model can produce solutions that would intuitively not be considered optimal. However, if the need for a highly workable mixture was paramount than the model has and can provide mixtures for each scenario that consider such a possibility.

3.6.4.2. Case 3: Scenario 1C – Interior column

A 16" by 16" limitation on column size and utilization of the parameters in **Table 3-16** resulted in the set of mixture design solutions shown in **Figure 3-28**. Application of a constant boundary condition for this case investigation resulted in cover being represented as a single point on the parallel axis plot. Red solution lines

indicate results from Scenario 1C while the blue lines are those from the baseline, Scenario 0.

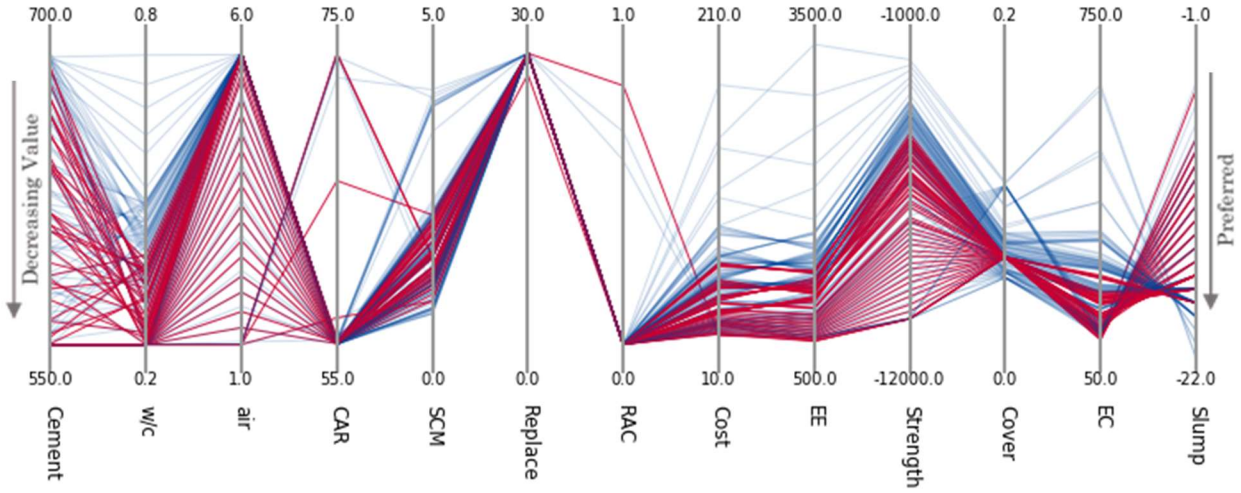

Figure 3-28. Solution set for the interior column

As anticipated, limiting the column size resulted in a reduced solution space from the baseline. Strength is reported starting at 4,400 psi rather than 1,800 psi seen in the baseline, indicating that the column size constraint is indeed constraining solutions and preventing mixtures that would result in smaller columns from being chosen. Results also validate the application of the constraint, since the cost, embodied energy, and embodied carbon are all below the average result values from Scenario 0. As seen through other functional units in this study, the solutions consist exclusively of fly ash and silica fume SCM mixtures with replacements at, or near, 30%. Additionally, results contain similar high variability in cement content and air content, although w/c ratio diversity is depressed. Results show that only one mixture utilizes recycled aggregates, although no chloride exposure is considered. The presence of only one recycled aggregate containing solution parallels previous cases with no chloride exposure that had few or no

recycled aggregate containing solutions. Again, this mixture produces the lowest possible cost. Slump is reduced from previous results with the minimum value at 3.8 cm or 1.5 inches, which is attributable to the reduced w/c ratio options.

3.6.4.3. Case 3: Scenario 2C – Exterior bridge pier

The results of the bridge pier designed to be larger than 36" by 36" and implemented in the harsh environment specified in **Table 3-16** are presented in **Figure 3-29.** Solutions for Scenario 2C are plotted in blue without comparison to Scenario 0 solutions.

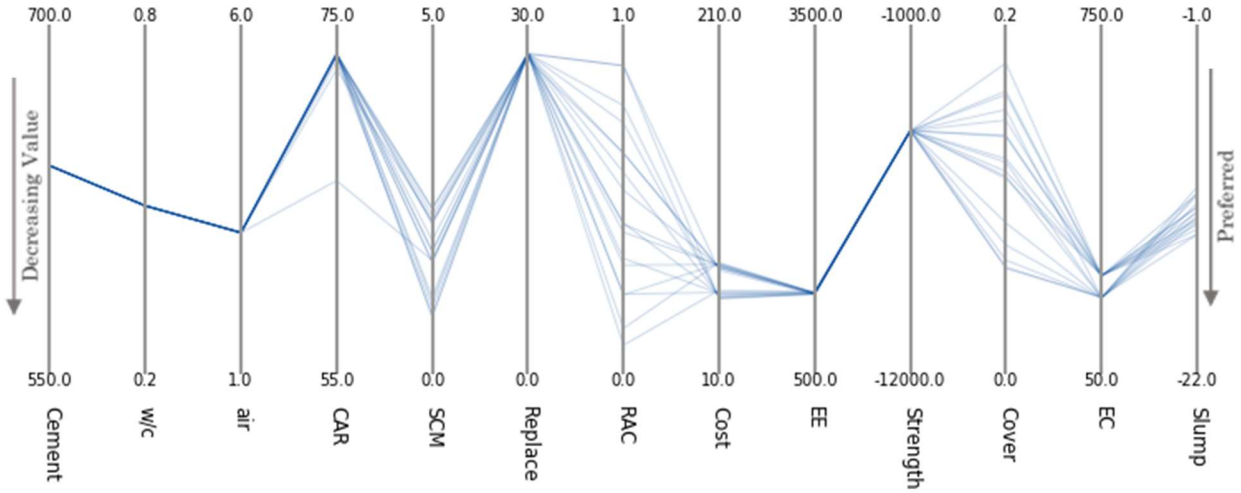

Figure 3-29. Solution set for exterior bridge pier

With such extreme constraints, the solution set is reduced from the baseline and constitutes only 16 solutions compared to 160. Results are only varied for 6 of the 13 variables with cement content, w/c ratio, air content, coarse aggregate content, SCM replacement, embodied energy and strength found to be nearly fixed for all solutions. SCM choices are restricted to silica fume and fly ash, as seen previously. However, the results do present viable ranges of recycled aggregate contents. These results are considered optimal, even though the bridge pier is in the

harshest chloride exposure environment considered in the model. Such results indicate that recycled aggregates may be more likely to be used by the model to explore a very confined solution space. Attributable to the varied use of recycled aggregates, cover varies by over 100% in these results and indicates, that in sufficiently large volumes of concrete, such as 36" by 36" pier, cover has little impact on the resulting volume of concrete and thus does not drive the optimization model. The fixed value of all solutions for embodied energy is surprising, given that these solutions do not share similar costs or embodied carbon.

3.7. CONCLUSIONS

This study presented the development, calibration, and implementation of a multi-objective concrete mixture design optimization tool that leveraged the Borg MOEA algorithm to overcome existing limitations of optimized concrete mixture design. The model considers six diverse and conflicting objective functions, namely compressive strength, workability, cost, embodied energy, embodied carbon, and chloride diffusivity. These functions were defined by empirical, statistical, and numerical relationships that were used to predict the values of the six considered design performance criteria. Additionally, a novel 1D numerical diffusion model developed to predict chloride diffusion in recycled aggregate concrete and a new model for carbon sequestration were included in the objective function formulation. Utilizing an evolutionary algorithm search-based methodology that relies on epsilon dominance on-dominant Pareto-optimal mixture design solutions were obtained for ten scenarios, resulting in a total of 17 solution sets across three case study examples, namely a cubic meter of concrete, a concrete tilt-up wall, and an axially loaded concrete column. These cases addressed a continuum of economic, spatial, and practical design criteria with a goal of demonstrating the capabilities of the MOEA framework by elucidating the tradeoffs between objectives and decisions.

Results confirm the suitability of the Borg MOEA algorithm for continued use with concrete mixture design optimization, due not only to the ease of implementation, but also to the diversity of solutions produced under conflict-inducing application criteria. Through the three case study examples, variations in number and type of constraint and objective function definitions were evident in the resulting solution sets. Additionally, variation in input parameters resulted in solution set variation, illustrating interesting trade-offs. With the flexibility and computational speed of the Borg utilized in this study, additional objective and

decision inclusion in the problem formulation can be provided in the further. Such prospects further validate the prospect of MOEA use for concrete optimization.

The results of seventeen scenario investigations illustrate the diversity and range of a solution set is proportional to the problem complexity, with the simplest application, Case 1, a cubic meter of concrete, producing the largest and most diverse solution fields. Increasingly complicated Case 2, the concrete tilt-up wall, and Case 3, the axially loaded concrete column, resulted in reduced sizes of solution sets. The final scenario (Case 3), a bridge pier, resulted in a solution set that was reduced to 16 mixtures from 160 in the comparatively simplistic column model baseline. Results also illustrate that transportation coefficients and constituent coefficients interact based on their relative magnitude. If a constituent is defined with small values for its impact metrics of cost, embodied carbon, and embodied energy, then transportation is more likely to dominate the solution field of a mixture containing that constituent. Fly ash mixtures were shown to exhibit this behavior more frequently with global changes to transportation impact parameters most affecting their solution set. However, on average, solution sets were found to consist of approximately 80% material impacts and 20% transportation impacts, indicating that subsequent and compounding transportation manipulations result in reduced tradeoffs with each manipulation. In regard to the inclusion of recycled aggregates, results showed that the use of recycled aggregates in structural applications in high-chloride environments is viable. In these investigations, recycled aggregates produce low-cost concretes and were only viable solutions in applications not requiring chloride resistance or when readily available (closer and cheaper) than their virgin counter parts. This finding is in agreement with previous research [17] that showed how exposure, rather than contamination of aggregates or their replacement percentage, was the primary factor affecting performance.

The results from this study further indicate that all ranges of cement content, w/c ratio, and air content can exist in optimal solutions in most solution sets without strength or workability constraints. Even for those applications requiring moderate chloride durability, water to cement ratios as high as 80%, which reduce diffusion coefficients, can appear in the solution set. Additionally, mixtures containing high cement replacements were found herein to be preferred when low environmental impact solutions are desired. In these cases, in particular, the importance of SCMs in determining the variability of solutions chosen by the model is most evident. This finding is in agreement with current practice in concrete mixture formulation. Again, the functionality of the model for the investigated cases and the variability in results indicates not only the viability of the Borg MOEA for concrete mixture design, but also the ability of the current model formulation to holistically capture trends found in current mixture design and research.

CHAPTER 4 SUMMARY

This thesis presented two studies that address the complexity of concrete mixture design, which is a multi-variable problem. Substantial time and effort goes into designing, testing, and perfecting single mixture designs for optimal performance in some, but not all, of these properties. This work showed that one constituent in particular, namely recycled aggregates, can have a drastic effect on concrete durability. Subsequently, a model was formulated to optimize concrete mixtures with considerations for mixture cost and embodied impacts, such as energy and carbon. By considering complex relationships between concrete mixture proportions and resulting constituent properties these two studies were used to help elucidate trade-offs and illustrate mixture design choices that may not have been considered by the current mixture designer.

The first study presented the development, validation, and implementation of a 1D numerical service-life prediction model for reinforced recycled aggregate concrete exposed to internal and external sources of chlorides. The model accounted for the inclusion of supplementary cementitious constituents (SCMs), namely (a) fly ash, (b) slag, (c) silica fume, and (d) metakaolin, and recycled aggregates (i) with and (ii) without initial chloride contamination from previous in-service exposure. The model was used to predict time to corrosion-induced cracking for reinforced recycled aggregate concrete in five case-study applications, namely structures in a marine splash zone (Zone I), a marine spray zone (Zone II), within 800 km of coastline (Zone III), within 1.5 km of coastline (Zone IV), and parking structures at locations greater than 1.5 km from the coastline (Zone V) in Los Angeles, California

and Anchorage, Alaska. The effects of recycled aggregate size, replacement ratio, degree of aggregate pre-contamination with chloride from previous in-service exposure, water-to-cement (w/c) ratio, and SCMs on time-to-cracking of reinforced recycled aggregate concrete were elucidated herein. The potential for SCMs to improve the service life of recycled aggregate concrete was investigated by estimating additions required to meet a target service life of 50 years.

Results indicate that, in addition to geographic location, temperature, and severity of exposure, w/c ratio and aggregate replacement ratio exhibited the greatest impact on time to chloride-induced cracking in reinforced recycled aggregate concrete. Furthermore, initial aggregate chloride contamination and aggregate size imparted minimal effects on expected service life. Finally, the results illustrate that the use of either fly ash or slag was most viable in achieving a 50-year service life for recycled aggregate concretes in chloride-laden environments.

Broadening the work of the first study, the second study presented the development and implementation of a multi-objective concrete mixture design tool capable of evaluating tradeoffs of different mixture proportions on concrete performance. The model was driven by a multi-objective evolutionary algorithm (MOEA) which uses a search-based methodology to find a set of Pareto-optimal mixture designs. Relationships informing the MOEA consider cement content, water, supplementary cementitious constituents (SCMs), namely (i) fly ash, (ii) silica fume, (iii) slag and (iv) metakaolin, sand, coarse aggregate, recycled aggregates, and air in the mixture. Six objectives functions were used to determine optimality of mixtures: strength, workability, chloride induced corrosion resistance, embodied energy, embodied carbon, and cost. Objective properties were modeled using a suite of empirical and numerical methods, which considered multiple decisions in their formulation. As demonstrated, the model can produce a suite of

optimal mixtures for three case study applications: a cubic meter of concrete, a tilt-up concrete wall, and a concrete column. In total 17 scenarios were investigated to illustrate the capabilities of the modeling methodology.

Results indicate that the model can elucidate tradeoffs related to SCM choice, recycled aggregate content, transportation cost, constituent cost, and application. Additionally, the results illustrate that practical, realistic design constraints, including a mixture of durability and strength criteria, can be implemented in the model and produce a set of viable, optimal, yet varied, solutions. Finally, the results obtained with this model suggest the need for continued refinement of the relationships that inform the optimization process, as well as the inclusion of additional objectives and decisions, to push the model even nearer to the goal of a comprehensive, holistic design methodology.

4.1. RECOMMENDATIONS FOR FUTURE WORK

Many practice constructions, such as a concrete wall, sidewalk, etc., can be simplified as a two-dimensional shape, which can be accurately modeled with the 1D diffusion model. Currently the proposed model of **Chapter 3** utilized two 1D diffusion models that were well suited for tasks such as those case investigations presented. However, although a column functional unit is included in this study the implementation of the 1D diffusion solution to predict durability for a 3D object results in an introduced solution error. A better approach to implement in the model would be to enhance the 1D diffusion model to a 2D model that could analyze 3D objects: beams, columns, foundations, etc. With the added dimension of analysis comes an increase in computational expense, as it would require, at minimum, an additional loop to solve for cover depth.

Mixtures in industry often include admixtures (which were excluded from the model formulation) to improve sub-par properties of a mixture design. Superplasticizers, for example, in particular are often utilized to improve workability for low w/c ratios (below 30%) as well as concretes including high SCM additions, which greatly reduce fresh state workability. While chemical superplasticizers often have significant embodied impacts, their inclusion can be beneficial when high workability and low w/c are desired. Currently this work does not address the contribution that admixtures can impart to a solution set. Rather, current implementation simply allows for bounds to be placed on w/c ratio, SCM replacement, and other decisions to allow for feasible solutions.

Empirical relationships, while simple to implement, are limited in their scope. Consequently, models that employ such empirical relationships are limited by their shortcomings. Often, data sets for concrete mixtures exist that consider more input variables and prediction outputs than existing empirical relationships. To take advantage of these data, sets artificial neural networks (ANN) could be applied within the model and trained using experimental data to derive more complete relationships for desired properties. Through the combination of similar and complementary data sets, ANN could be more effectively utilized and gaps in previous methods could be better addressed. Alternatively, existing research has indicated the potential for inclusion of stochastic sampling techniques and ANN to simulate data sets to further inform additional neural network models ([77], [83], [106], [111]). Implementing a prediction methodology that accounts for all decision variables, or as many of them as the most complete method allows, will lead to a more holistic mixture design methodology than is currently possible. Such a method could most immediately be utilized to produce a relationship for strength that accounts for cement, w/c ratio, SCM type and replacement, aggregate gradation,

admixture addition, air content, and recycled aggregates, and would address one of the greatest limitations of the current modeling approach.

Life cycle inventory (LCI) data, embodied carbon and embodied energy coefficients, inform two of the six objective functions. With such a large influence on the search procedure, high quality and accurate LCI data needs to be ensured within the model to produce practical trade-offs. Currently, no unified, comprehensive and public LCI database exists for use with this model. Additionally, United States specific LCI data is much more difficult to find than that of European or Canadian products. The model proposed in Chapter 3 utilizes data that took significant amounts of time to find and validate, ensuring confidence in results. If, more complete LCI databases were to be combined with the model then confidence in results would be much less time consuming to ensure.

REFERENCES

All references for this thesis are presented in order as they appear within the text.

[1] P. Mehta and P. Monteiro, *Concrete microstructure, properties and materials*. NY: McGraw-Hill, 2003.

[2] J. de Brito and N. Saikia, *Recycled aggregate in concrete: use of industrial, construction and demolition waste*. London ; New York: Springer, 2013.

[3] M. Malešev, V. Radonjanin, and S. Marinković, "Recycled Concrete as Aggregate for Structural Concrete Production," *Sustainability*, vol. 2, no. 5, pp. 1204–1225, Apr. 2010.

[4] M. L. Berndt, "Properties of sustainable concrete containing fly ash, slag and recycled concrete aggregate," *Constr. Build. Mater.*, vol. 23, no. 7, pp. 2606–2613, Jul. 2009.

[5] K.-H. Yang, Y.-B. Jung, M.-S. Cho, and S.-H. Tae, "Effect of supplementary cementitious materials on reduction of CO2 emissions from concrete," *J. Clean. Prod.*, vol. 103, pp. 774–783, Sep. 2015.

[6] A. Petek Gursel, E. Masanet, A. Horvath, and A. Stadel, "Life-cycle inventory analysis of concrete production: A critical review," *Cem. Concr. Compos.*, vol. 51, pp. 38–48, Aug. 2014.

[7] F. Pacheco Torgal, S. Miraldo, J. A. Labrincha, and J. De Brito, "An overview on concrete carbonation in the context of eco-efficient construction: Evaluation, use of SCMs and/or RAC," *Constr. Build. Mater.*, vol. 36, pp. 141–150, Nov. 2012.

[8] M. C. G. Juenger and R. Siddique, “Recent advances in understanding the role of supplementary cementitious materials in concrete,” *Cem. Concr. Res.*, vol. 78, pp. 71–80, Dec. 2015.

[9] V. G. Papadakis, “Effect of supplementary cementing materials on concrete resistance against carbonation and chloride ingress,” *Cem. Concr. Res.*, vol. 30, no. 2, pp. 291–299, Feb. 2000.

[10] K. A. Riding, M. D. A. Thomas, and K. J. Folliard, “Apparent Diffusivity Model for Concrete Containing Supplementary Cementitious Materials,” *ACI Mater. J.*, vol. 110, no. 6, pp. 705–713, 2013.

[11] S.-C. Kou and C.-S. Poon, “Long-term mechanical and durability properties of recycled aggregate concrete prepared with the incorporation of fly ash,” *Cem. Concr. Compos.*, vol. 37, pp. 12–19, Mar. 2013.

[12] N. Otsuki, S. Miyazato, and W. Yodsudjai, “Influence of Recycled Aggregate on Interfacial Transition Zone, Strength, Chloride Penetration and Carbonation of Concrete,” *J. Mater. Civ. Eng.*, vol. 15, no. 5, pp. 443–451, 2003.

[13] B. Hu, B. Liu, and L. Zhang, “Chloride ion permeability test and analysis for recycled concrete,” *J. Hefei Univ. Technol.*, vol. 32, no. 8, pp. 1240–3, 2009.

[14] Y. A. Villagrán-Zaccardi, C. J. Zega, and Á. A. Di Maio, “Chloride Penetration and Binding in Recycled Concrete,” *J. Mater. Civ. Eng.*, vol. 20, no. 6, pp. 449–455, 2008.

[15] R. Movassaghi, “Durability of Reinforced Concrete Incorporating Recycled Concrete as Aggregate(RCA),” University of Waterloo, Mechanical and Mechatronics Engineering, 2006.

[16] F. Debieb, L. Courard, S. Kenai, and R. Degeimbre, “Mechanical and durability properties of concrete using contaminated recycled aggregates,” *Cem. Concr. Compos.*, vol. 32, no. 6, pp. 421–426, Jul. 2010.

[17] W. V. Srubar, “Stochastic service-life modeling of chloride-induced corrosion in recycled-aggregate concrete,” *Cem. Concr. Compos.*, vol. 55, pp. 103–111, Jan. 2015.

[18] Life-365 Consortium III, *Life-365 service life prediction model and computer program for predicting the service life and life-cycle cost of reinforced concrete exposed to chlorides*. 2014.

[19] K. Henchi, E. Samson, F. Chapdelaine, and J. Marchand, “Advanced Finite-Element Predictive Model for the Service Life Prediction of Concrete Infrastructures in Support of Asset Management and Decision-Making,” 2007, pp. 870–880.

[20] J. Xiao, J. Ying, and L. Shen, “FEM simulation of chloride diffusion in modeled recycled aggregate concrete,” *Constr. Build. Mater.*, vol. 29, pp. 12–23, Apr. 2012.

[21] J. Ying, J. Xiao, and V. W. Y. Tam, “On the variability of chloride diffusion in modelled recycled aggregate concrete,” *Constr. Build. Mater.*, vol. 41, pp. 732–741, Apr. 2013.

[22] K. Tuutti, “Corrosion of steel in concrete,” Swedish Cement and Concrete Research Institute, CBI Research Report 4–82, 1982.

[23] K. Hong and R. D. Hooton, “Effects of cyclic chloride exposure on penetration of concrete cover,” *Cem. Concr. Res.*, vol. 29, no. 9, pp. 1379–1386, Sep. 1999.

[24] J. Crank, *The mathematics of diffusion*, 2. ed., Reprinted. Oxford: Oxford Univ. Press, 2009.

[25] M. D. A. Thomas and P. B. Bamforth, “Modelling chloride diffusion in concrete,” *Cem. Concr. Res.*, vol. 29, no. 4, pp. 487–495, Apr. 1999.

[26] R. Duval and E. . Kadri, “Influence of Silica Fume on the Workability and the Compressive Strength of High-Performance Concretes,” *Cem. Concr. Res.*, vol. 28, no. 4, pp. 533–547, Apr. 1998.

[27] S. Wild, J. M. Khatib, and A. Jones, “Relative strength, pozzolanic activity and cement hydration in superplasticised metakaolin concrete,” *Cem. Concr. Res.*, vol. 26, no. 10, pp. 1537–1544, Oct. 1996.

[28] L. Lam, Y. . Wong, and C. . Poon, “Effect of Fly Ash and Silica Fume on Compressive and Fracture Behaviors of Concrete,” *Cem. Concr. Res.*, vol. 28, no. 2, pp. 271–283, Feb. 1998.

[29] K. H. Obla, “Specifying Fly Ash for Use in Concrete,” *Concr. Focus*, pp. 60–66, 2008.

[30] N. S. Ismael and M. N. Ghanim, “Properties of blended cement using metakaolin and hydrated lime,” *Adv. Cem. Res.*, vol. 27, no. 6, pp. 321–328, Jun. 2015.

[31] Portland Cement Association (PCA), “Fly Ash, Slag, Silica Fume, and Natural Pozzolans,” in *Design and Control of Concrete Mixtures*, 2002, pp. 57–72.

[32] C.-M. Aldea, F. Young, K. Wang, and S. P. Shah, “Effects of curing conditions on properties of concrete using slag replacement,” *Cem. Concr. Res.*, vol. 30, no. 3, pp. 465–472, Mar. 2000.

[33] T. Liu and R. . Weyers, “Modeling the Dynamic Corrosion Process in Chloride Contaminated Concrete Structures,” *Cem. Concr. Res.*, vol. 28, no. 3, pp. 365–379, Mar. 1998.

[34] T. El Maaddawy and K. Soudki, “A model for prediction of time from corrosion initiation to corrosion cracking,” *Cem. Concr. Compos.*, vol. 29, no. 3, pp. 168–175, Mar. 2007.

[35] D. V. Val and L. Chernin, “Cover cracking in reinforced concrete elements due to corrosion,” *Struct. Infrastruct. Eng.*, vol. 8, no. 6, pp. 569–581, Jun. 2012.

[36] I. Balafas and C. J. Burgoyne, “Environmental effects on cover cracking due to corrosion,” *Cem. Concr. Res.*, vol. 40, no. 9, pp. 1429–1440, Sep. 2010.

[37] T. J. Kirkpatrick, R. E. Weyers, M. M. Sprinkel, and C. M. Anderson-Cook, “Impact of specification changes on chloride-induced corrosion service life of bridge decks,” *Cem. Concr. Res.*, vol. 32, no. 8, pp. 1189–1197, Aug. 2002.

[38] Y. Tanaka, H. Kawano, H. Watanabe, and T. Nakajo, “Study on cover depth for prestessed conceret bridges in airborne-chloride environments,” *PCI J.*, vol. 51, no. 2, pp. 42–53, 2006.

[39] J. Zhang and Z. Lounis, “Sensitivity analysis of simplified diffusion-based corrosion initiation model of concrete structures exposed to chlorides,” *Cem. Concr. Res.*, vol. 36, no. 7, pp. 1312–1323, Jul. 2006.

[40] Z. Lounis, “Uncertainty Modeling of Chloride Contamination and Corrosion of Concrete Bridges,” in *Applied Research in Uncertainty Modeling and Analysis*, vol. 20, N. O. Attoh-Okine and B. M. Ayyub, Eds. Boston, MA: Springer US, 2005, pp. 491–511.

[41] G. K. Glass and N. R. Buenfeld, “The presentation of the chloride threshold level for corrosion of steel in concrete,” *Corros. Sci.*, vol. 39, no. 5, pp. 1001–1013, May 1997.

[42] C. W. Li, “Life-Cycle Modeling of Corrosion-Affected Concrete Structures: Propagation,” *J. Struct. Eng.*, vol. 129, no. 6, pp. 753–761, 2003.

[43] C. Q. Li, "Reliability Based Service Life Prediction of Corrosion Affected Concrete Structures," *J. Struct. Eng.*, vol. 130, no. 10, pp. 1570–1577, 2004.

[44] S. Caré, Q. T. Nguyen, V. L'Hostis, and Y. Berthaud, "Mechanical properties of the rust layer induced by impressed current method in reinforced mortar," *Cem. Concr. Res.*, vol. 38, no. 8–9, pp. 1079–1091, Aug. 2008.

[45] M. Etxeberria, E. Vázquez, A. Marí, and M. Barra, "Influence of amount of recycled coarse aggregates and production process on properties of recycled aggregate concrete," *Cem. Concr. Res.*, vol. 37, no. 5, pp. 735–742, May 2007.

[46] F. Debieb, L. Courard, S. Kenai, and R. Degeimbre, "Roller compacted concrete with contaminated recycled aggregates," *Constr. Build. Mater.*, vol. 23, no. 11, pp. 3382–3387, Nov. 2009.

[47] Portland Cement Association (PCA), "Aggregates for Concrete," in *Design and control of concrete mixtures*, pp. 79–103.

[48] J. Xiao, J. Li, and C. Zhang, "Mechanical properties of recycled aggregate concrete under uniaxial loading," *Cem. Concr. Res.*, vol. 35, no. 6, pp. 1187–1194, Jun. 2005.

[49] S. M. Levy and P. Helene, "Durability of recycled aggregates concrete: a safe way to sustainable development," *Cem. Concr. Res.*, vol. 34, no. 11, pp. 1975–1980, Nov. 2004.

[50] S. Goto and D. M. Roy, "The effect of w/c ratio and curing temperature on the permeability of hardened cement paste," *Cem. Concr. Res.*, vol. 11, no. 4, pp. 575–579, Jul. 1981.

[51] V. Corinaldesi and G. Moriconi, "Influence of mineral additions on the performance of 100% recycled aggregate concrete," *Constr. Build. Mater.*, vol. 23, no. 8, pp. 2869–2876, Aug. 2009.

[52] K. Y. Ann, H. Y. Moon, Y. B. Kim, and J. Ryou, "Durability of recycled aggregate concrete using pozzolanic materials," *Waste Manag.*, vol. 28, no. 6, pp. 993–999, 2008.

[53] B. Esmaeilkhanian, K. H. Khayat, and O. H. Wallevik, "Mix design approach for low-powder self-consolidating concrete: Eco-SCC—content optimization and performance," *Mater. Struct.*, vol. 50, no. 2, p. 124, Apr. 2017.

[54] T. H. Kim, S. H. Tae, S. J. Suk, G. Ford, and K. H. Yang, "An Optimization System for Concrete Life Cycle Cost and Related CO2 Emissions," *Sustainability*, vol. 8, no. 4, p. 361, Apr. 2016.

[55] R. Liu, S. A. Durham, K. L. Rens, and A. Ramaswami, "Optimization of Cementitious Material Content for Sustainable Concrete Mixtures," *J. Mater. Civ. Eng.*, vol. 24, no. 6, pp. 745–753, Jun. 2012.

[56] D. Hadka and P. Reed, "Borg: An Auto-Adaptive Many-Objective Evolutionary Computing Framework," *Evol. Comput.*, vol. 21, no. 2, pp. 231–259, May 2013.

[57] R. Smith, J. Kasprzyk, and E. Zagona, "Many-Objective Analysis to Optimize Pumping and Releases in Multireservoir Water Supply Network," *J. Water Resour. Plan. Manag.*, vol. 142, no. 2, p. 04015049, Feb. 2016.

[58] J. R. Kasprzyk, P. M. Reed, and D. M. Hadka, "Battling Arrow's Paradox to Discover Robust Water Management Alternatives," *J. Water Resour. Plan. Manag.*, vol. 142, no. 2, p. 04015053, Feb. 2016.

[59] S. Gopinath, A. Murthy, D. Ramya, and N. Iyer, "Optimised mix design for normal strength and high performance concrete using particle packing method," *Arch. Civ. Eng. Wars.*, vol. LVII, no. 4, p. 357, Dec. 2011.

[60] T. Blankendaal, P. Schuur, and H. Voordijk, "Reducing the environmental impact of concrete and asphalt: a scenario approach," *J. Clean. Prod.*, vol. 66, pp. 27–36, Mar. 2014.

[61] P. Van den Heede and N. De Belie, "Environmental impact and life cycle assessment (LCA) of traditional and 'green' concretes: Literature review and theoretical calculations," *Cem. Concr. Compos.*, vol. 34, no. 4, pp. 431–442, Apr. 2012.

[62] ACI Committe 211, "Standard practive for slecting proportions for normal, heavyweight and mass concrete." American Concrete Institute, USA, 1991, reapproved in-2009.

[63] A. Singh and K. Gautam, "Comparison of ISI and ACI methods for absolute volume concrete mix design," *30th Conf. OUR WORLD Concr. Struct.*, pp. 399–406, Aug. 2005.

[64] K. C. Hover, "Graphical Approach to mixture proportioning by ACI 211.1-91," *Concr. Int.*, pp. 49–53, 1995.

[65] Portland Cement Association (PCA), "Designing and Proportioning Normal Concrete Mixtures." EB001, Ch. 9; p. 149-177.

[66] ISO 14040:1997(E), "Environmental management -- Life cycle assessment -- Principles and framework." Committee ISO/TC 207/SC 5.

[67] W. Meng, M. Valipour, and K. H. Khayat, "Optimization and performance of cost-effective ultra-high performance concrete," *Mater. Struct.*, vol. 50, no. 1, p. 29, Feb. 2017.

[68] I.-C. Yeh, "Computer-aided design for optimum concrete mixtures," *Cem. Concr. Compos.*, vol. 29, no. 3, pp. 193–202, Mar. 2007.

[69] S. Ahmad and S. A. Alghamdi, "A Statistical Approach to Optimizing Concrete Mixture Design," *Sci. World J. Cairo*, 2014.

[70] S. Ahmad, “optimum concrete mixture design using locally available ingredients,” *Arab. J. Sci. Eng.*, vol. 32, no. 1, pp. 27–33.

[71] K. A. Soudki, E. F. El-Salakawy, and N. B. Elkum, “Full Factorial Optimization of Concrete Mix Design for Hot Climates,” *J. Mater. Civ. Eng.*, vol. 13, no. 6, 2001.

[72] M. . Alves, R. . Cremonini, and D. C. . Dal Molin, “A comparison of mix proportioning methods for high-strength concrete,” *Cem. Concr. Compos.*, vol. 26, no. 6, pp. 613–621, Aug. 2004.

[73] P. L. J. Domone and M. Soutsos, “An approach to the proportioning of high strength concrete mixes,” *Concr. Int.*, vol. 6, no. 10, pp. 26–31, 1994.

[74] P. Goltermann, V. Johansen, and L. Palbøl, “Packing of Aggregates: An Alternative Tool to Determine the optimal aggregate Mix,” *ACI Mater. J.*, vol. 94, no. 5, pp. 435–443, 1997.

[75] A. Amirjanov and K. Sobolev, “Optimization of a Computer Simulation Model for Packing of Concrete Aggregates,” *Part. Sci. Technol.*, vol. 26, no. 4, pp. 380–395, Jul. 2008.

[76] B. E. Jimma and P. R. Rangaraju, “Chemical admixtures dose optimization in pervious concrete paste selection – A statistical approach,” *Constr. Build. Mater.*, vol. 101, Part 1, pp. 1047–1058, Dec. 2015.

[77] S. Kostić, N. Vasović, and B. Marinković, “Robust optimization of concrete strength estimation using response surface methodology and Monte Carlo simulation,” *Eng. Optim.*, vol. 49, no. 5, pp. 864–877, May 2017.

[78] M. Berry, B. Kappes, and L. Kappes, “Optimization of Concrete Mixtures Containing Reclaimed Asphalt Pavement,” *ACI Mater. J. Farmington Hills*, vol. 112, no. 6, pp. 723–733, Dec. 2015.

[79] M. Muthukumar, D. Mohan, and M. Rajendran, “Optimization of mix proportions of mineral aggregates using Box Behnken design of experiments,” *Cem. Concr. Compos.*, vol. 25, no. 7, pp. 751–758, Oct. 2003.

[80] O. Akalin, K. U. Akay, and B. Sennaroglu, “Self-Consolidating High-Strength Concrete Optimization by Mixture Design Method,” *ACI Mater. J. Farmington Hills*, vol. 107, no. 4, pp. 357–364, Aug. 2010.

[81] E. Güneyisi, M. Gesoğlu, Z. Algın, and K. Mermerdaş, “Optimization of concrete mixture with hybrid blends of metakaolin and fly ash using response surface method,” *Compos. Part B Eng.*, vol. 60, pp. 707–715, Apr. 2014.

[82] N. Bouzoubaâ and B. Fournier, “Optimization of fly ash content in concrete,” *Cem. Concr. Res.*, vol. 33, no. 7, pp. 1029–1037, Jul. 2003.

[83] I.-C. Yeh, “Modeling slump flow of concrete using second-order regressions and artificial neural networks,” *Cem. Concr. Compos.*, vol. 29, no. 6, pp. 474–480, Jul. 2007.

[84] A. Bilgil, “Estimation of slump value and Bingham parameters of fresh concrete mixture composition with artificial neural network modelling,” *Sci. Res. Essays*, vol. 5, no. 8, pp. 1753–1765, 2012.

[85] R. Siddique, P. Aggarwal, and Y. Aggarwal, “Prediction of compressive strength of self-compacting concrete containing bottom ash using artificial neural networks,” *Adv. Eng. Softw.*, vol. 42, no. 10, pp. 780–786, Oct. 2011.

[86] M. H. Rafiei, W. H. Khushefati, R. Demirboga, and H. Adeli, “Neural Network, Machine Learning, and Evolutionary Approaches for Concrete Material Characterization,” *ACI Mater. J.*, vol. 113, no. 6, Dec. 2016.

[87] C.-H. Lim, Y.-S. Yoon, and J.-H. Kim, “Genetic algorithm in mix proportioning of high-performance concrete,” *Cem. Concr. Res.*, vol. 34, no. 3, pp. 409–420, Mar. 2004.

[88] I.-C. Yeh, “Optimization of concrete mix proportioning using a flattened simplex–centroid mixture design and neural networks,” *Eng. Comput.*, vol. 25, no. 2, p. 179, Mar. 2009.

[89] A. Yadollahi, E. Nazemi, A. Zolfaghari, and A. M. Ajorloo, “Optimization of thermal neutron shield concrete mixture using artificial neural network,” *Nucl. Eng. Des.*, vol. 305, pp. 146–155, Aug. 2016.

[90] B. Şimşek, Y. T. İç, and E. H. Şimşek, “A TOPSIS-based Taguchi optimization to determine optimal mixture proportions of the high strength self-compacting concrete,” *Chemom. Intell. Lab. Syst.*, vol. 125, pp. 18–32, Jun. 2013.

[91] M.-Y. Cheng, D. Prayogo, and Y.-W. Wu, “Novel Genetic Algorithm-Based Evolutionary Support Vector Machine for Optimizain High-Performance Concrete Mixture,” *J Comput Ov Eng*, vol. 28, no. 4, pp. 1–7, 2014.

[92] M. Liu, “The Research and Simulation of PID Algorithm Optimization for Concrete Mixing,” *Appl. Mech. Mater. Zurich*, vol. 602–605, pp. 1198–1201, Aug. 2014.

[93] B. Ahmadi-Nedushan, “An optimized instance based learning algorithm for estimation of compressive strength of concrete,” *Eng. Appl. Artif. Intell.*, vol. 25, no. 5, pp. 1073–1081, Aug. 2012.

[94] S. A. Alghamdi, S. Ahmad, and A. Lawan, “Optimization of Concrete Mixture Design and Cover Thickness for Reinforced Concrete Members under Chloride Exposure,” *ACI Mater. J. Farmington Hills*, vol. 113, no. 5, pp. 589–598, Oct. 2016.

[95] M. A. Jayaram, M. C. Nataraja, and C. N. Ravikumar, “Elitist Genetic Algorithm Models: Optimization of High Performance Concrete Mixes,” *Mater. Manuf. Process.*, vol. 24, no. 2, pp. 225–229, Jan. 2009.

[96] T. Ji, Y. Yang, M. Fu, B. Chen, and H.-C. Wu, “Optimum Design of Reactive Powder Concrete Mixture Proportion Based on Artificial Neural and Harmony Search Algorithm,” *ACI Mater. J. Farmington Hills*, vol. 114, no. 1, pp. 41–47, Feb. 2017.

[97] J.-H. Lee, Y.-S. Yoon, and J.-H. Kim, “A new heuristic algorithm for mix design of high-performance concrete,” *KSCE J. Civ. Eng.*, vol. 16, no. 6, pp. 974–979, Sep. 2012.

[98] C. A. Coello Coello, G. B. Lamont, and D. A. Van Veldhuizen, *Evolutionary algorithms for solving multi-objective problems*, 2. ed. New York, NY: Springer, 2007.

[99] P. M. Reed, D. Hadka, J. D. Herman, J. R. Kasprzyk, and J. B. Kollat, “Evolutionary multiobjective optimization in water resources: The past, present, and future,” *Adv. Water Resour.*, vol. 51, pp. 438–456, Jan. 2013.

[100] D. Hadka and P. Reed, “Diagnostic Assessment of Search Controls and Failure Modes in Many-Objective Evolutionary Optimization,” *Evol. Comput.*, vol. 20, no. 3, pp. 423–452, Sep. 2012.

[101] M. Laumanns, L. Thiele, K. Deb, and E. Zitzler, “Combining Convergence and Diversity in Evolutionary Multiobjective Optimization,” *Evol. Comput.*, vol. 10, no. 3, pp. 263–282, Sep. 2002.

[102] J. B. Kollat and P. M. Reed, “A computational scaling analysis of multiobjective evolutionary algorithms in long-term groundwater monitoring applications,” *Adv. Water Resour.*, vol. 30, no. 3, pp. 408–419, Mar. 2007.

[103] S. Popovics, “Analysis of the concrete strength versus water-cement ratio relationship,” *ACI Mater. J.*, vol. 87, no. 5, pp. 517–529, 1990.

[104] M. F.M. Zain and S. M. Abd, "Multiple Regression Model for Compressive Strength Prediction of High Performance Concrete," *J. Appl. Sci.*, vol. 9, no. 1, pp. 155–160, Jan. 2009.

[105] K. Jankovic, D. Nikolic, D. Bojovic, L. Loncar, and Z. Romakov, "The estimation of compressive strength of normal and recycled aggregate concrete," *Facta Univ. - Ser. Archit. Civ. Eng.*, vol. 9, no. 3, pp. 419–431, 2011.

[106] S. Chithra, S. R. R. S. Kumar, K. Chinnaraju, and F. Alfin Ashmita, "A comparative study on the compressive strength prediction models for High Performance Concrete containing nano silica and copper slag using regression analysis and Artificial Neural Networks," *Constr. Build. Mater.*, vol. 114, pp. 528–535, Jul. 2016.

[107] C. F. Ferraris, "Measurement of the rheological properties of high performance concrete: State of the art report," *J. Res. Natl. Inst. Stand. Technol.*, vol. 104, no. 5, p. 461, Sep. 1999.

[108] R. J. Flatt, "Towards a prediction of superplasticized concrete rheology," *Mater. Struct.*, vol. 37, no. 5, pp. 289–300, Jun. 2004.

[109] M. Nehdi and M.-A. Rahman, "Estimating rheological properties of cement pastes using various rheological models for different test geometry, gap and surface friction," *Cem. Concr. Res.*, vol. 34, no. 11, pp. 1993–2007, Nov. 2004.

[110] C. K. Park, M. H. Noh, and T. H. Park, "Rheological properties of cementitious materials containing mineral admixtures," *Cem. Concr. Res.*, vol. 35, no. 5, pp. 842–849, May 2005.

[111] V. Chandwani, V. Agrawal, and R. Nagar, "Modeling slump of ready mix concrete using genetic algorithms assisted training of Artificial Neural Networks," *Expert Syst. Appl.*, vol. 42, no. 2, pp. 885–893, Feb. 2015.

[112] N.-D. Hoang and A.-D. Pham, "Estimating Concrete Workability Based on Slump Test with Least Squares Support Vector Regression," *J. Constr. Eng.*, vol. 2016, p. e5089683, Dec. 2016.

[113] EPA, "Inventory of U.S. Greenhouse Gas Emissions and Sinks: 1990-2012," U.S. Environmental Protection Agency, Office of Atmospheric Programs, Washington, D.C., EPA 430-R-11-005, 2014.

[114] ACAA, "2012 Coal Combustion Product (CCP) Prodution and Use Survey Report," American Coal Ash Association, Aurora, CO, 2012.

[115] Robert Snow Means Company, *Building Costruction Cost Data*, Version 2016. 1996.

[116] R. Jones, M. McCarthy, and M. Newlands, "Fly Ash Route to Low Embodied CO2 and Implications for Concrete Construction," presented at the World of Coal Ash Conference, Denver, CO, 2011.

[117] WARM Version 13, "Fly Ash EPD," 2015.

[118] ASTM, "EPD for slag cement," Slag Cement Association, 2014.

[119] J. R. Prusinski and M. G. Van Geem, "Life Cycle Inventory of Slag Cement Concrete," presented at the Eighth CANMET/ACI International Conference on Fly Ash, Silica Fume, Slag and Natural Pozzolans in Concrete, 2006.

[120] H. Aysha, T. García-Segura, N. Arunachalam, A. Ramachandra Murthy, and N. R. Iyer, "Assessment of Embodied Energy in the Production of Ultra High Performance Concrete (UHPC)," *Int. J. Stud. Res. Technol. Manag.*, vol. 2, no. 3, p. 113, 120AD.

[121] K. Scrivener and A. Favier, *Calcined Clays for Sustainable Concrete: Proceedings of the 1st International Conference on Calcined Clays for Sustainable Concrete*. Springer Netherlands, 2015.

[122] Vulcan Material Company Western Division, “Environmental Product Declaration for 12 concrete aggregate products,” Climate Earth, Inc., 2016.

[123] Holcim, “EPD of Aggregates,” Holcim Romania, Declaration number: S-P-00528, 2014.

[124] M. Wijayasundara, R. H. Crawford, and P. Mendis, “Comparative assessment of embodied energy of recycled aggregate concrete,” *J. Clean. Prod.*

[125] A. Horvath, “Life-Cycle Environmental and Economic Assessment of Using Recycled Materials for Asphalt Pavements,” 2003.

[126] Oak Ridge National Laboratory, “Transportation Energy Data Book Edition 35.” U.S. Department of Energy, 2016.

[127] “Emission Factors for Greenhouse Gas Inventories.” U.S. Environmental Protection Agency, Apr-2014.

[128] F. . Torrey and D. Murray, “An analysis of the Operational Costs of Trucking: A 2014 Update.” American Transportation Research Institute, 2014.

[129] Association of American Railroads, “Average U.S. Freight rail rates.” .

[130] M. L. Marceau, M. A. Nisbet, and M. G. VanGeem, “Life Cycle Inverntory of Portland Cement Concrete.” Portland Cement Association, 2007.

[131] NRMCA, “NRMCA member industry-wide EPD for ready mix concrete.” EPD10046, 2014.

[132] A. Steffens, D. Dinkler, and H. Ahrens, “Modeling carbonation for corrosion risk prediction of concrete structures,” *Cem. Concr. Res.*, vol. 32, no. 6, pp. 935–941, Jun. 2002.

[133] S. K. Roy, K. B. Poh, and D. o. Northwood, “Durability of concrete—accelerated carbonation and weathering studies,” *Build. Environ.*, vol. 34, no. 5, pp. 597–606, Sep. 1999.

[134] W. Ashraf, “Carbonation of cement-based materials: Challenges and opportunities,” *Constr. Build. Mater.*, vol. 120, pp. 558–570, Sep. 2016.

[135] C. Pade and M. Guimaraes, “The CO2 uptake of concrete in a 100 year perspective,” *Cem. Concr. Res.*, vol. 37, no. 9, pp. 1348–1356, Sep. 2007.

[136] T. García-Segura, V. Yepes, and J. Alcalá, “Life cycle greenhouse gas emissions of blended cement concrete including carbonation and durability,” *Int. J. Life Cycle Assess.*, vol. 19, no. 1, pp. 3–12, Jan. 2014.

[137] A. Souto-Matrinez, E. A. Delesky, K. E. O. Foster, and W. V. Srubar, “A Mathematical Model for predicting the Carbon Sequestration Potential of Ordinary Portland Cement (OPC) Concrete,” *Cem. Concr. Res.*, vol. In review, 2016.

[138] I. Monteiro, F. A. Branco, J. de Brito, and R. Neves, “Statistical analysis of the carbonation coefficient in open air concrete structures,” *Constr. Build. Mater.*, vol. 29, pp. 263–269, Apr. 2012.

APPENDIX – A: RECYCLED CONCRETE MODELING CODE

A.1 MASTER

```
% Nathan Stambaugh

clc; clear; close all

filepath = sprintf('%s%s',cd,'\');
mail = 'eralman@gmail.com';
password = 'yamatonds';
host = 'smtp.gmail.com';
port = '465';

setpref( 'Internet','E_mail', mail );
setpref( 'Internet', 'SMTP_Server', host );
setpref( 'Internet', 'SMTP_Username', mail );
setpref( 'Internet', 'SMTP_Password', password );

props = java.lang.System.getProperties;
props.setProperty( 'mail.smtp.user', mail );
props.setProperty( 'mail.smtp.host', host );
props.setProperty( 'mail.smtp.port', port );
props.setProperty( 'mail.smtp.starttls.enable', 'true' );
props.setProperty( 'mail.smtp.debug', 'true' );
props.setProperty( 'mail.smtp.auth', 'true' );
props.setProperty( 'mail.smtp.socketFactory.port', port );
props.setProperty( 'mail.smtp.socketFactory.class', 'javax.net.ssl.SSLSocketFactory' );
props.setProperty( 'mail.smtp.socketFactory.fallback', 'false' );

%% Inputs that effect simulation speed and accuracy
ns = 1; % number of Monte-Carlo simulations
maxlife = 150; % max service life in years
slices = 300; %Number of slices through the concrete

%% INPUTS TO INVESTIGATE
mix = 2; % 1 = OPC, 2 = SF, 3 = FA, 4 = SG; 5 = MK;
City = 'LA'; % LA or AK
number = 6.0; % number >= 6 means default values

[mca,stdca,w2c,sigma,AR_Ratio] = SIMS(number);
```

```
% Overwrite values from above function
% mca = 0; stdca = 0.0;% kg/m^3 initial contamination of agg
sigmam = convlength(sigma,'in','m'); % diam of each aggregate (TRY: 3/8, 1/2, 1)

concwt = 2350; % concrete weight (lb/m^3)
DRMAint = 12.5e-12; % m^2/sec
cover = convlength(2.5,'in','m'); % cover depth (TRY: 2, 2.5, 3)
DiamReb = convlength(0.375,'in','m');
mci = 0.7; stdci = 0.05;%kg/m^3 Chloride threshold of rebar

if City == 'LA';
% average temp in each month for Los Angles CA in (K) Jan-Dec
Temp = [286.95;287.2;287.55;288.75;290.25;291.85;293.75;294.55;294.25;292.45;289.55;286.95];
elseif City == 'AK';
% average temp in each month for Anchorage AL in (K) Jan-Dec
Temp = [264.8;266.8;270.1;275.7;282.0;286.2;287.8;287.0;282.3;274.5;267.9;265.9];
end
Tref = 21; % (C) reference tempurature
Tref = Tref+273.15; % Tref in (K)

for q = 1:5
Location = q; % Condition to be investigated
% 1 = Marine Splash Zone; 2 = Marine Spray Zone; 3 = Within 800m of Coast
% 4 = Within 1.5km of Coast; 5 = Further than 1.5 km from Coast
% for w/c = 0.45 average for locations:
% 1 ~ 5, 2 ~ 9, 3 ~ 13, 4 ~ 18, 5 ~ 45

%% Calculations

SL = maxlife*365; % days in max service life

% days in each month from Jan-Dec
months = [31;28;31;30;31;30;31;31;30;31;30;31];

% Transformation of tempurature vector from above
t = repmat(months,maxlife,1); % builds an vector of each year by months
t = cumsum(t,1); % adds each previous entry to the next cumulatively suming
T = repmat(Temp,maxlife,1); % vector of tempuratures of length time
% this array is Kelvin and the time goes till the max service life.

% Converts the mean and standard deviation of the initiation limit into
% lognormal values.
varci = stdci^2;% kg/m^2 varriance of the chloride initiation threshold
% below are values needed to create log normal distribtion
muci = log((mci)/sqrt(1+varci/(mci^2)));
sigci = sqrt(log(varci/(mci^2)+1));

depth = cover*2; % depth of analysis
spacemin = cover/10;
num_agg = floor(AR_Ratio*(cover-spacemin)/(sigmam)); % num agg in the cover depth
x_s = ((cover/num_agg)- sigmam); % space each agg can be placed in
```

```
num_agg = 1:num_agg;
dx = depth/slices; % slice thickness
slice_c = round(cover/dx); % slices in cover thickness

dt = 1; % step size of analysis in months

%% Boundary Condition Generation
[Max_conc,t2max,bc] = Boundary(maxlife,Location,concwt);

%% Initial Diffusion Coefficents
D28 = (2.17*10^-12)*exp(w2c/0.279); % m^2/sec

%% Mix Design Function

[D,DRAMT] = mixdesign(mix,t,T,D28,DRMAint,Tref,SL);

%% Time to Cracking and Initiation Variables
for i = 1:ns
[Tc(i)] = Cracking(DiamReb,cover);
   ci(i) = lognrnd(muci,sigci);
   ca(i) = normrnd(mca,stdca);
end

%% Time to Failure

[Tfail] = Time2Fail(ns,slices,t,dt,dx,D28,ci,ca,D,DRAMT,DRMAint,num_agg,slice_c,sigmam,Tc,bc);

FileName = sprintf('%s Location %1.0f SCM %1.0f mca=%1.2f w2c=%1.2f sig=%1.2f
ARR=%1.2f',City,Location,mix,mca,w2c,sigma,AR_Ratio);
save(sprintf('%s%s',FileName,'.mat'),'Tfail');
%sendmail(mail,'hello from the otherside');
sendmail('eralman@gmail.com','Data','Dont Keep',{sprintf('%s%s%s',filepath,FileName,'.mat')});
end
msgbox({'Operation Completed' '' 'IF YOU ARE READING THIS THEN PLEASE CLOSE OUT OF
MATLAB AND THE COMPUTER IS ALL YOURS'},'User','warn');
```

A.2 SIMS

```
function [mca,stdca,w2c,sigma,AR_Ratio] = SIMS(number)
% Default Values for all sims
mca = 2; stdca = 0.01; w2c = 0.45; sigma = 0.375; AR_Ratio = 0.5;


if number < 2;
   % Initial Contamination
   if number == 1.0;
      mca = 1.0; stdca = 0.01;
   elseif number == 1.1;
      mca = 1.5; stdca = 0.01;
   elseif number == 1.2;
      mca = 2.0; stdca = 0.01;
   elseif number ==1.3;
      mca = 2.5; stdca = 0.01;
   end

elseif number < 3;
   %Water Cement Ratio
   if number == 2.0;
      w2c = 0.3;
   elseif number == 2.1;
      w2c = 0.35;
   elseif number == 2.2;
      w2c = 0.40;
   elseif number ==2.3;
      w2c = 0.45;
   end

elseif number < 4;
   % Aggregate Size
   if number == 3.0;
      sigma = 0.375;
   elseif number == 3.1;
      sigma = 0.5;
   elseif number == 3.2;
      sigma = 0.75;
   elseif number == 3.3;
      sigma = 1;
   end

elseif number < 5;
   % Recycled Agg Replacement Ratio
   if number == 4.0;
      AR_Ratio = 0.3;
```

```
    elseif number == 4.1;
        AR_Ratio = 0.5;
    elseif number == 4.2;
        AR_Ratio = 0.7;
    elseif number == 4.3;
        AR_Ratio = 1;
    end

elseif number < 6;
    % No Recycled Aggregate
    AR_Ratio =0;
    sigma = 0;
    w2c = 0.45;
end
```

A.3 MIX DESIGN

```
function [D,DRMAT] =  mixdesign(mix,t,Temp,D28,DRMA,Tref,SL)



% The ultimate diffusion coefficent is found for 100 years of service
U = 35000;
R = 8.314462;
SF = 5; % Fixed input for Silica Fume
FA = 5; % Fixed input for Fly Ash
SG = 5; % Fixed input for Slag
MK = 5; % Fixed input for Metakoalin

if mix == 1; % 100% OPC
    m = 0.26; %Decay of diffusion coefficent
    Dult = D28*(28/SL)^m; % m^2/sec; The ultimate diffusion coefficent
    Dt = D28*(28./t).^m;
    Dt(SL*12/365+1:length(Dt)) = Dult;
    D = Dt.*exp((U/R)*(1/Tref-1./Temp)); % D(t,T)

elseif mix == 2; % Silica Fume
    m = 0.26; %Decay of diffusion coefficent
    D_SF = 0.206+0.794*exp(-SF/2.51);
    D28 = D28*D_SF;
    Dult = D28*(28/SL)^m; % m^2/sec; The ultimate diffusion coefficent
    Dt = D28*(28./t).^m;
    Dt(SL*12/365+1:length(Dt)) = Dult;
    D = Dt.*exp(U/R*(1/Tref-1./Temp)); % D(t,T)

elseif mix == 3; % Fly Ash
    m = 0.26+0.4*(FA/50); %Decay of diffusion coefficent
    D_FA = 0.171+0.829*exp(-FA/6.07);
    Dult = D28*(28/SL)^m; % m^2/sec; The ultimate diffusion coefficent
    Dt = D28*(28./t).^m;
    Dt = Dt.*D_FA;
    Dt(SL*12/365+1:length(Dt)) = Dult;
    D = Dt.*exp(U/R*(1/Tref-1./Temp)); % D(t,T)

elseif mix == 4; % Slag
    m = 0.26+0.4*(SG/70); %Decay of diffusion coefficent
    Dult = D28*(28/SL)^m; % m^2/sec; The ultimate diffusion coefficent
    Dt = D28*(28./t).^m;
    Dt(SL*12/365+1:length(Dt)) = Dult;
    D = Dt.*exp(U/R*(1/Tref-1./Temp)); % D(t,T)

elseif mix == 5; % Metakaolin
```

```
    m = 0.26;
    D_MK = 0.191+0.809*exp(-MK/6.12);
    Dult = D28*(28/SL)^m; % m^2/sec; The ultimate diffusion coefficent
    D28 = D28*D_MK;
    Dt = D28*(28./t).^m;
    Dt(SL*12/365+1:length(Dt)) = Dult;
    D = Dt.*exp(U/R*(1/Tref-1./Temp)); % D(t,T)
end

% Recycled Aggregate Concrete Degredation
DRMAT = DRMA*exp(U/R*(1/Tref-1./Temp)); % DRMA(T)
```

A.4 BOUNDARY

```
function [Max_conc, t2max, bc] = Boundary(maxlife,Location,concwt)

%% slopes for exposure conditions from Life 365
if Location == 1;
   Max_conc = concwt*.008;
   t2max = 1;
elseif Location == 2
   Max_conc = concwt*.01;
   t2max = 10;
elseif Location == 3
   Max_conc = concwt*0.006;
   t2max = 15;
elseif Location == 4
   Max_conc = concwt*0.006;
   t2max = 30;
else
   Max_conc = concwt*0.008;
   t2max = 200;
end

%% the sloped portion of the boundary condition
if Max_conc == 0
   y = 0;
else
   y = linspace(0,Max_conc,t2max*12);
end
y = y.';

%% the constant portion of the boundary condition
if maxlife <= t2max
   yy = repmat(Max_conc, 1, 1);
else
   yy = repmat(Max_conc, abs((maxlife-t2max))*12,1);
end
bc = [y; yy];
```

A.5 CRACKING

```
function [Tc] = Cracking(DiamReb,cover)

% all values below assumed for general concrete mix.
fts.m = 544; fts.s = 72.5; % Tensile strength (psi)
Em.m = 4350000; Em.s = 435000; % Modulus of elasticity (psi)
tpr.m = 0.0005; tpr.s = 0.0002; % porous region thickness (ft)
iCorr.m = 0.0015; iCorr.s = 0.0002; % Corrosion rate (A/ft^2)
Phi = 2; % Creep coefficent of Concrete
v = 0.28; % Poisson's Ratio
rho_r = 225; % Density of rust
rho_s = 490; % Density of steel
alpha_l = 0.523; % lower bound on alpha
alpha_h = 0.622; % upper bound on alpha
DiamReb = convlength(DiamReb,'m','in');
cover = convlength(cover,'m','in');

ft = abs(normrnd(fts.m,fts.s));
E = abs(normrnd(Em.m,Em.s));
tp = abs(normrnd(tpr.m,tpr.s));
alpha = abs(alpha_l+(alpha_h-alpha_l)*rand);
icorr = abs(normrnd(iCorr.m,iCorr.s));

E_eff = E/(1+Phi);
a = (DiamReb+2*tp)/2;
b = cover+a;
tcrit = cover*ft/E_eff*((a^2+b^2)/(b^2-a^2)+v);
Wporous = pi*rho_r*32.14*tp/12*(DiamReb/12);
Wexpand = pi*rho_r*32.14*(DiamReb/12+2*tp/12)*tcrit/12;

Wcrit = rho_s/(rho_s-alpha*rho_r)*(Wporous+Wexpand);
kp = (1/alpha)*pi*DiamReb*icorr;
Tc = (Wcrit^2)/(2*kp);

end
```

A.6 TIME TO FAIL

```
function [Tfail,Ti] =
Time2Fail(ns,slices,t,dt,dx,D28,ci,ca,D,DRAMT,DRMAint,num_agg,slice_c,sigma,Tc,bc)
Ti = zeros(1,ns);
Tfail = zeros(1,ns);
for i = 1:ns
   r = zeros(numel(t),slices);
   D_c = zeros(slices,slices); % Diffusion coefficient time t => slice s
   flag = zeros(slices,1); % flag for agg locations
   A = zeros(slices,slices); % A matrix
   B = zeros(slices,slices); % B matrix
   C = zeros(slices,1); % Concentration at time t+1 slice s
   c = zeros(slices,1); % Concentration at time t slice s
   UU = zeros(slices,numel(t));
   if num_agg > 0;
      % start coordinate of each aggregate placed randomly
      xagg_s = ceil(num_agg/num_agg(1)*((slice_c-
(numel(num_agg)*sigma/dx))/numel(num_agg))*ceil(20*rand())/20 + sigma*num_agg/dx -
sigma*num_agg(1)/dx);
      % end coordinate of each aggregate placed randomly
      xagg_e = round(xagg_s+sigma/dx-1);
      for a = 1:length(num_agg)
         flag(xagg_s(a):(xagg_e(a)-1)) = 1;
      end
   end

   % initial condiations and diffusion with time for the whole thickness
   % of analysis
   for j = 1:numel(t)
      for s = 1:slices
         if flag(s) == 1;
            if j == 1;
               D_c(1,s) = DRMAint;
               c(s) = ca(i);
            else
               D_c(j,s) = DRAMT(j);
            end
         else
            if j == 1;
               D_c(1,s) = D28;
               c(s) = 0;
            else
               D_c(j,s) = D(j);
            end
         end
      end
   end
```

```
    for j = 1:numel(t)
        for k = 1:slices
            r(j,k) = D_c(j,k)*60*60*24*30*(dt)/(2*(dx^2));
            if k == 1 || k == slices
                A(k,k) = 1;
                B(k,k) = 1;
            else
                A(k,k) = 1+2*r(j,k); A(k,k+1) = -r(j,k); A(k,k-1) = -r(j,k);
                B(k,k) = 1-2*r(j,k); B(k,k+1) = r(j,k); B(k,k-1) = r(j,k);
            end
        end
        c(1) = bc(j);
        C = A\B*c;
        UU(:,j) = c;
        c = C;
        if j == length(t); % reached max service life
            Ti(i) = t(j)/365; % time to initiation in years
            %fprintf('found a ti = %3.0f for number for sim = %5.0f \n',maxlife,i);
            break
        elseif c(slice_c) > ci(i);
            Ti(i) = t(j)/365;
            %fprintf('found a ti < %3.0f number for sim = %5.0f \n',maxlife,i);
            break
        end

    end
    Tfail(i) = Tc(i)+Ti(i);
end
end
```

APPENDIX – B: CONCRETE MIXTURE OPTIMIZATION CODE

B.1 WRAPPER MODULE

The wrapper module is responsible for connecting all other modules and is what calls the Borg algorithm through the use of a command initiated within a sub process. This script also contains the macro levers that control the model, such as service life, chloride exposure, functional unit type, service life model type.

```
__author__ = 'nast1697'
import time
start = time.time()

print("You started Case #1")

# My Python files:
from problemmodules import *
from wrappermodules import *

# Other Python modules
import subprocess
from collections import OrderedDict

# USER-DEFINED PARAMETERS THAT DON'T CHANGE BETWEEN RUNS

borgExecutableName = "borgPC.exe"  # NS - this has been edited from ./borgPC.exe to run on Nate's computers

seed = 1
nfe = 10000
pythonName = "mix_for_borg_M3Mod1.py"

# In the following dictionaries, we want to preserve the order we declare things in,
# since we are going to be interfacing with a 'dumb' simulation model that expects
# to see the variables in the same order every time.
# http://stackoverflow.com/questions/1867861/python-dictionary-keep-keys-values-in-same-order-as-declared

# List objectives here
objectives = OrderedDict()
objectives["cost"] = Objective("cost", 1.0)
objectives["ee"] = Objective("ee", 5.0)
objectives["strength"] = Objective("strength", 100.0)
objectives["cover"] = Objective("cover", 0.005)
# NS - added embodided carbon objective (Kg CO2e)
objectives["ec"] = Objective("ec", 1.0)
# NS - added workability objective (cm of slump)
```

```
objectives["workability"] = Objective("workability", 1.0)
num_objectives = len(objectives)

# List decisions here
decisions = OrderedDict()
decisions["cement"] = Decision("cement", 550.00, 700.00)
decisions["wc"] = Decision("wc", 0.25, 0.75)
decisions["air"] = Decision("air", 1.00, 6.00)
decisions["ca"] = Decision("ca", 55.00, 75.00)
# NS - added SCM numbers greater than 3.49 so slag can be an exclusive choice
decisions["scm"] = Decision("scm", 0.50, 4.49)
decisions["replace"] = Decision("replace", 0.00, 30.00)
# NS - new decision of recycled aggregate content (% of total aggregate)
decisions["rca"] = Decision("rca", 0.0, 1.00)
num_decisions = len(decisions)

# NS - This is needed because the number of constraints must be defined
num_constraints = 1

# case = "exterior_column"
case = 5
name = "Baseline_sequesteredC02"
service_life = 25
# NS - use chloride_exposure as 5.0, 7.0, etc when running the simple diffusion model. Otherwise use integer from
1 - 5
chloride_exposure = 3.0

service_model = 1.0  # 0.0 = constant cover; 1.0 = 1D; all other = simplified
functional_unit = 2.0  # 0.0 = m^3; 1.0 = Column; all other = 1D tilt-up wall

filetype = ".csv"
# NS - Created a function that will build the file name so it doesnt have to be edited each time
resultFileName = result_file_name(name,service_life, chloride_exposure, service_model, functional_unit,
filetype)

pythonArgs = '{:.2f} {:.2f} {:.2f} {:.2f}'.format(service_life, chloride_exposure, service_model, functional_unit)

# Builds the system command that will call the borgMOEA and run the mix_for_borg in a subprocess
systemCommand = construct_system_command(borgExecutableName, resultFileName, seed, nfe,
num_constraints, objectives, decisions, pythonName, pythonArgs)

run_algorithm = 0
if run_algorithm:
   print "Running M^3 Model #1"
   subprocess.call(systemCommand, shell=True)
else:
   print "Algorithm not running. Program assumes that baseline have already been created"

# NS -  Function that can run the monte carlo sim if desired
runMonte = 0  # NS - binary run variable
monteName = "MonteCarlo"
monteFiletype = ".csv"
numSims = 1024

monteCarloSim = monte_carlo_sim(runMonte, numSims, monteName, service_life, chloride_exposure,
monteFiletype,
                        num_objectives, decisions, num_decisions, num_constraints)
```

```
# NS - this parallel plotting plots all objectives and decisions (13 in total)
plot1 = parallel_plot_all(resultFileName, objectives, decisions)

plot = plotting_3D(resultFileName, objectives, decisions)

end = time.time()
print("All done in %0.7f seconds" % (end - start))
```

B.2 MIX FOR BORG MODULE

The Borg call module is responsible for communicating between the Borg and concrete modules. It also contains the property values for many of the constituents used in the model.

```
__author__ = 'josephkasprzyk and nast1697'
# My Python files:
from concretemodules import *

# Other Python modules
from sys import *
import argparse

# https://docs.python.org/2/howto/argparse.html#id1
parser = argparse.ArgumentParser(description='Process model parameters.')
parser.add_argument("service_life", type=float, help="Service life for use in calculations")
parser.add_argument("chloride_exposure", type=float, help="The chloride exposure")
parser.add_argument("service_model", type=float, help = "The service life model choice")
parser.add_argument("functional_unit", type=float, help = "The functional unit choice")
args = parser.parse_args()
service_life = args.service_life
chloride_exposure = args.chloride_exposure
serv_model = args.service_model
func_unit = args.functional_unit

while True:
    # Read the next line from standard input
    line = raw_input()

    # Stop if the Borg MOEA is finished
    if line == "":
        break

    # Parse the decision variables from the input
    vars = map(float, line.split())

    # Evaluate the problem

    # Decision variables:
    # cement content, w_c_ratio, air_content, c_a_ratio_percent, r_slag, r_choice, r_scm_replace, r_agg_contnent

    # Note that the upper and lower bounds in the comments below are just suggestions.  The real bounds
    # are set in the call to borg.exe

    cement_content = vars[0]          # range: 550 to 700
    w_c_ratio = vars[1]               # range: 0.25 to 0.75
    air_content = vars[2]             # range: 1 to 3
    c_a_ratio_percent = vars[3]       # range: 55 to 75
    # r_slag = vars[4]                # range: 0 to 50
    r_scm_choice = int(round(vars[4]))  #range: 0.5 to 4.5, rounded to 1, 2, 3 or 4
    r_scm_replace = vars[5]           # range: 0 to 30,
```

```
    r_agg_content = vars[6]          # range: 0 to 1

    # NS - Added slag to the SCM list because it wass being used every mix otherwise
    if r_scm_choice == 1:
        r_fly_ash = r_scm_replace
        r_silica_fume = 0.0
        r_metakaolin = 0.0
        r_slag = 0.0
    elif r_scm_choice == 2:
        r_fly_ash = 0.0
        r_silica_fume = r_scm_replace
        r_metakaolin = 0.0
        r_slag = 0.0
    elif r_scm_choice == 3:
        r_fly_ash = 0.0
        r_silica_fume = 0.0
        r_metakaolin = r_scm_replace
        r_slag = 0.0
    elif r_scm_choice == 4:
        r_fly_ash = 0.0
        r_silica_fume = 0.0
        r_metakaolin = 0.0
        r_slag = r_scm_replace

    # Arguments: Name, cost ($/kg-mi), embodied energy (MJ/kg-mi), embodied carbon (kgCO2e/kg-mi)
    components = dict()
    components["transportation"] = Component("transportation", 0.0000091, 0.00034, 0.00031)

    # Arguments: Name, cost, embodied energy, embodied carbon, distance, specific gravity
    constituents = dict()
    constituents["coarse agg"] = Constituent("coarse aggregate", 0.021, 0.1, 0.0061, 50.0, 2.5)
    # NS - Added recycled aggregate as a constituent
    constituents["recycled agg"] = Constituent("recycled aggregate", 0.01, 0.05, 0.003, 10.0, 2.6)
    constituents["fine agg"] = Constituent("fine aggregate", 0.023, 0.08, 0.0076, 75.0, 2.63)
    constituents["cement"] = Constituent("cement", 0.17, 5.9, 0.9, 50.0, 3.15)
    constituents["water"] = Constituent("water", 0.005, 0.01, 0.0001, 20.0, 1.0)

    # Arguments: name, cost, embodied energy, embodied carbon, distance, specific gravity, replacement
percentage, beta
    scms = dict()
    scms["slag"] = SCM("slag", 0.106, 1.6, 0.146, 100.0, 2.9, r_slag, 0.38)
    # NS - used fly ash beta for class C because more prevalent
    scms["fly ash"] = SCM("fly ash", 0.065, 0.1, 0.01, 100.0, 2.4, r_fly_ash, 0.27)
    scms["silica fume"] = SCM("silica fume", 0.44, 0.04, 0.7, 100.0, 2.25, r_silica_fume, 0.99)
    scms["metakaolin"] = SCM("metakaolin", 0.36, 2.08, 0.6, 100.0, 2.5, r_metakaolin, 0.55)

    ## NS - SERVICE LIFE MODEL CALL!

    # NS - Arguments: service life, chloride exposure, chloride rebar threshold, cover depth lower (m), cover depth
upper (m),
    # service life model (where 1.0 = 1D, 0.0 = no model, and all others is a steady state model)
    # serv_model = 1.0
    serv_param = ServiceLifeParameters(service_life, chloride_exposure, 0.7, 0.02, 0.2, serv_model)

    # NS - This is the temperature vector for LA
    temperature = [286.95, 287.2, 287.55, 288.75, 290.25, 291.85, 293.75, 294.55, 294.25, 292.45, 289.55, 286.95]

    # NS - Arguments: number of number of slices (integer), time step (months, integer), diameter of rebar (in),
```

```
    # depth of analysis (m), recycled_agg replace (decimal), recylced_agg_diffusion (m^2/s), agg_contamination,
agg_size (m)
    d_rma = 12.5 * 10 ** -12

    one_D_model_param = SLModel1D(250, 12, 0.75, 0.2, r_agg_content, d_rma, 0.5, 0.012)

    ## NS - FUNCTIONAL UNIT CALL!
    # NS - Arguments: name, monthly temperature (K), Exposure(1.0 - 5.0 integers) which is controlled in the
Basecase script through the chloride_exposure
    exposure_class = chloride_exposure
    # Functional unit of 0.0 is m^3; 1.0 is a column, all others are a m by m tilt-up wall
    # func_unit = 2.0
    location_param = Site Location("LA", temperature, exposure_class, func_unit)

    ## NS - SEQUESTRATION MODEL CALL!

    # Alpha dictionary for sequestration model
    alpha = dict()
    alpha["Type I"] = 0.165
    alpha["Type II"] = 0.163
    alpha["Type III"] = 0.166
    alpha["Type IV"] = 0.135
    alpha["Type V"] = 0.161
    alpha["Type White"] = 0.203

    # Exposure dictionary for sequestration model
    carbonexpo = dict()
    carbonexpo["XC1"] = CarbonExpo(1.0, 0.0)
    carbonexpo["XC2"] = CarbonExpo(0.2, 0.183)
    carbonexpo["XC3"] = CarbonExpo(0.77, 0.02)
    carbonexpo["XC4"] = CarbonExpo(0.41, 0.085)

    # NS - Arguments: Cement Type (Type I, II, III, IV, V; White), Alpha Dictionary, CO2 Concentration (ppm), k0,
k2,
    # Exposure Coefficient Dictionary, Exposure class (XC1, XC2, XC3, XC4), degree of carbonation (0-1.0), model
control (binary)
    sequestration_param = CarbonSequestration("Type I", alpha, 300.0, 3.0, 1.0, carbonexpo, "XC1", 1.0, 0.0)

    ## NS - MIX DESIGN MODEL CALL!

    # Arguments: cement content, w/c ratio, air content percentage, c/a ratio percentage,
    # python objects: components, constituents, scms, serv_param, stoc_model_param
    mymix = MixDesign(cement_content, w_c_ratio, air_content, c_a_ratio_percent, components, constituents,
scms, serv_param, one_D_model_param, location_param, sequestration_param)
    mymix.calc_absolute_volume()

    # NS - Because new functions were created to run the 1D service model they must be called by the borg to work
    if serv_model == 1.0:
        mymix.calc_cracking_time()
        mymix.calc_boundary_condition()
    if serv_model > 0.0:
        mymix.diffusion_coefficient()

    mymix.calc_agg_placement()
    mymix.calc_service_life()
    mymix.calc_func_unit()
    mymix.calc_sequestered_carbon()
```

```
    # NS -  These objective constraints should be hard coded in the same order as the definitions of the objectives

    objsconstrs = [mymix.func_vol_total_cost, mymix.func_vol_embodied_energy, (-
1.0)*mymix.compressive_strength,
                   mymix.cover_depth, mymix.func_vol_embodied_carbon, (-1.0)*mymix.workability,
mymix.strength_constr]

    # Print objectives to standard output, flush to write immediately
    print " ".join("%0.10f" % val for val in objsconstrs)
    stdout.flush()
```

B.3 WRAPPER MODULES

The wrapper modules consist of the class and function definitions required to produce a file name, call the Borg and call the parallel plotting script

```
__author__ = 'joka0958 and nast1697'


class Objective(object):
    def __init__(self, name, epsilon):
        self.name = name
        self.epsilon = epsilon


class Decision(object):
    def __init__(self, name, lower, upper):
        self.name = name
        self.lower = lower
        self.upper = upper


# NS - Created function for the file name
# NS - This can be and is used for both the monte-carlo and MOEA outputs


def result_file_name(name, servlife, clexp, servmod, funcunit, filetype):
    # This function is used to construct the output file names for the program
    # Example: tradeoff_s25_c5.txt

    # The first component is the desired file name
    resultFileName = name

    # The service life
    resultFileName = resultFileName + "_sl%d" % servlife

    # The chloride exposure
    resultFileName = resultFileName + "_c%d" % clexp

    # The service life model
    resultFileName = resultFileName + "_slm%d" % servmod

    # The functional unit
    resultFileName = resultFileName + "_fu%d" % funcunit

    # The file extension
    resultFileName = resultFileName + filetype

    return resultFileName


def construct_system_command(borgExecutableName, resultFileName, seed, nfe, num_constraints, objectives,
decisions, pythonName, pythonArgs):
    # This function constructs the borg system command.
    # Example:
```

```
# ./borg.exe -f cost-ee-strength-cover_highmiles_coarse_sl25_balanced_miles.txt
# -s 1 -v 7 -o 4 -e 0.5,2,50,0.005 -n 100000 -l 550.0,0.25,1.0,55.0,0.0,0.5,1.0
# -u 700.0,0.75,6.0,75.0,15.0,3.49,30.0 python mix_for_borg_B1.py

# The first component is the borg executable name
systemCommand = borgExecutableName

# The next several components can be in any order, but we will prescribe the order

# The result filename
systemCommand = systemCommand + " -f " + resultFileName

# The seed
systemCommand = systemCommand + " -s %d" % seed

# The run duration
systemCommand = systemCommand + " -n %d " % nfe

# Constraints
systemCommand = systemCommand + " -c %d " % num_constraints

# Objectives (two parts)

# Objectives Part 1: The number of objectives
systemCommand = systemCommand + " -o %d " % int(len(objectives))

# Objectives Part 2: The epsilons
systemCommand = systemCommand + " -e "

for key in objectives:
   systemCommand = systemCommand + "%.6f," % objectives[key].epsilon

systemCommand = systemCommand[:-1].strip() #get rid of the last comma

# The decisions (three parts)

# Decisions Part 1: The number of decisions
systemCommand = systemCommand + " -v %d " % int(len(decisions))

# Decisions Part 1: The lower bounds
systemCommand = systemCommand + " -l "

for key in decisions:
   systemCommand = systemCommand + "%.6f," % decisions[key].lower

systemCommand = systemCommand[:-1].strip() #get rid of the last comma

# Decisions Part 2: The upper bounds
systemCommand = systemCommand + " -u "

for key in decisions:
   systemCommand = systemCommand + "%.6f," % decisions[key].upper

systemCommand = systemCommand[:-1].strip() #get rid of the last comma

# Now the python part

systemCommand = systemCommand + " -- python " + pythonName + " " + pythonArgs
```

```
    return systemCommand

def construct_parallel_command(plotCodeName, plotFileName, dataFileName, columns, numcolumns,
precision, names, dataSetName, colorCode, minima, maxima, lineWidth):

    # Program that will excecute this code
    parallelCommand = "python "

    # Name of the code that plots the figure (parallel.py)
    parallelCommand = parallelCommand + plotCodeName

    # Output plot file name
    parallelCommand = parallelCommand + " " + plotFileName

    # File name where data is stored
    parallelCommand = parallelCommand + " " + dataFileName

    # NS - The remaining components can be called in any order but as with the system command we will specify
them
    #Columns whose data will be plotted from the file specified above
    parallelCommand = parallelCommand + " -C" + " %d-" %columns[0] + "%d" %columns[1]

    # Specify how many parallel axes the plot will have
    parallelCommand = parallelCommand + " -w %d" %numcolumns

    # data precision values for each column of data (variable) to be plotted
    parallelCommand = parallelCommand + " -p"

    for i in range(len(precision)):
        parallelCommand = parallelCommand + " %0.3f" %precision[i]

    parallelCommand = parallelCommand + " -a"

    # Axis name labels
    for i in range(len(names)):
        parallelCommand = parallelCommand + " %s" %names[i]

    # Name of the data set read in
    parallelCommand = parallelCommand + " -n " + dataSetName

    # Color code for the data
    parallelCommand = parallelCommand + " -c"

    for i in range(len(colorCode)):
        parallelCommand = parallelCommand + " %0.3f" %colorCode[i]

    # mimimum axis values for each variable, if not specified then will fit data
    parallelCommand = parallelCommand + " -m"

    for i in range(len(minima)):
        parallelCommand = parallelCommand + " %0.3f" %minima[i]

    # maximum axis values for each variable, if not specified then will fit data
    parallelCommand = parallelCommand + " -M"

    for i in range(len(maxima)):
        parallelCommand = parallelCommand + " %0.3f" %maxima[i]
```

```
# Line weight for output variable lines
parallelCommand = parallelCommand + " -W %0.1f" %lineWidth[0] + " %0.1f" %lineWidth[1]

return parallelCommand
```

B.4 CONCRETE MODULES

The concrete modules script is responsible for defining the classes of constituents and calculating all of the objectives. It contains the AVM function, service life function, functional unit function and sequestered carbon function amongst others.

```
__author__ = 'josephkasprzyk and nast1697'

# Other Python modules
from math import exp
from math import sqrt
from scipy import special

# NS - These imports are all needed for the 1D Stochastic Service Life model
from numpy import linspace
from numpy import ones
from numpy import zeros
from numpy import tile
from numpy import cumsum
from math import pi
from numpy import matmul
from numpy import linalg
from numpy import where
from numpy import concatenate
from math import floor
from math import ceil
from numpy import random


class Component(object):
    # A component is going to be any item that is in the cost or embodied energy calculation
    # including constituents (coarse agg, fine agg, cement, water, air), SCMs, and transport
    # NS - adding embodied carbon to the lis of properties
    def __init__(self, name, cost, embodied_energy, embodied_carbon):
        self.name = name
        self.cost = cost
        self.embodied_energy = embodied_energy
        self.embodied_carbon = embodied_carbon


class Constituent(Component):
    # Constituents also have a distance and specific gravity
    def __init__(self, name, cost, embodied_energy, embodied_carbon, distance, specific_gravity):
        Component.__init__(self, name, cost, embodied_energy, embodied_carbon)
        self.distance = distance
        self.specific_gravity = specific_gravity


class SCM(Constituent):
    # SCMs also have a replacement percentage and beta(sequestered carbon)
    def __init__(self, name, cost, embodied_energy, embodied_carbon, distance, specific_gravity,
```

```
replacement_percentage, beta):
        Constituent.__init__(self, name, cost, embodied_energy, embodied_carbon, distance, specific_gravity)
        self.replacement_percentage = replacement_percentage
        self.beta = beta

    def __str__(self):
        return "SCM %s with EE: %f, SG: %f" % (self.name, self.embodied_energy, self.specific_gravity)


class ServiceLifeParameters(object):
    def __init__(self, service_life, chloride_exposure, chloride_rebar_threshold, cover_depth_lower,
cover_depth_upper,
                 service_life_model):
        self.service_life = service_life
        self.chloride_exposure = chloride_exposure
        self.chloride_rebar_threshold = chloride_rebar_threshold
        self.cover_depth_lower = cover_depth_lower
        self.cover_depth_upper = cover_depth_upper
        # NS - create a service life model parameter that can call different service life model types e.g.
        # 1D - 2D models
        self.service_life_model = service_life_model

# NS - Created a new class of variables for the Stocastic 1D service life model


class SLModel1D(object):
    def __init__(self, slices, delta_time, diam_rebar, depth, ar_ratio, d_rma, rec_agg_c, agg_size):
        self.slices = slices
        self.delta_time = delta_time
        # NS - delta slice was removed because it is a function of cover depth and number of slices
        # self.delta_slice = delta_slice
        self.diam_rebar = diam_rebar
        self.depth = depth
        self.ar_ratio = ar_ratio
        self.d_rma = d_rma
        self.rec_agg_c = rec_agg_c
        self.agg_size = agg_size


# NS - Class for Location that can be used in service-life model 1D to determine temperature and exposure
classification
class SiteLocation(object):
    def __init__(self, name, temperature, exposure_class, func_unit):
        self.name = name
        self.temperature = temperature
        self.exposure_class = exposure_class
        self.func_unit = func_unit


class CarbonExpo(object):
    def __init__(self, k1, n):
        self.k1 = k1
        self.n = n


class CarbonSequestration(object):
    def __init__(self, cement_type, alpha, carbon_conc, k0, k2, carbon_expo, exposure, degree_carbonation,
model_control):
```

```
        self.cement_type = cement_type
        self.alpha = alpha
        self.carbon_conc = carbon_conc
        self.k0 = k0
        self.k2 = k2
        self.carbon_expo = carbon_expo
        self.exposure = exposure
        self.degree_carbonation = degree_carbonation
        self.model_control = model_control


class MixDesign(object):
    def __init__(self, cement_content, w_c_ratio, air_content_percent, c_a_ratio_percent, components,
constituents, scms,
                 service_life_parameters, model_1d, site_location, carbon_sequestration):
        self.cement_content = cement_content
        self.w_c_ratio = w_c_ratio
        self.air_content_percent = air_content_percent
        self.c_a_ratio_percent = c_a_ratio_percent
        self.components = components
        self.constituents = constituents
        self.scms = scms
        # This class of service_life_parameters is defined above
        self.service_life_parameters = service_life_parameters
        # This class of stocastic_model_1d parameters is defined above
        self.model_1d = model_1d
        self.site_location = site_location
        self.carbon_sequestration = carbon_sequestration

    def calc_absolute_volume(self):
        # all calculations in this method assume a unit volume of 1 square yard

        # WEIGHTS

        # calculate weights of the SCMs
        temp_weight = 0.0
        for key in self.scms:
            self.scms[key].abs_vol_weight = self.cement_content * self.scms[key].replacement_percentage / 100
            temp_weight = temp_weight + self.scms[key].abs_vol_weight

        # The weight of cement has been reduced by the SCMs
        self.constituents["cement"].abs_vol_weight = self.cement_content - temp_weight

        # The weight of water is determined by the w_c_ratio
        self.constituents["water"].abs_vol_weight = self.cement_content * self.w_c_ratio

        # VOLUMES

        # The volumes here are expressed in cubic feet, but we are assuming a volume of
        # 1 cubic yard, thus we need to multiply by 27 cubic feet per cubic yard:
        volume_conversion = 27.0

        # also we need the specific weight of water (lb per cubic ft), since the weights of constituents are
        # expressed using specific gravity
        specific_weight_water = 62.4

        # volume of air
        self.abs_vol_air = self.air_content_percent / 100.0 * volume_conversion
```

```
        # Now, to calculate the volume of the paste, we need to cycle through the SCMs and
        # determine the volume of each.

        temp_vol = 0.0
        for key in self.scms:
            self.scms[key].abs_vol_vol = self.scms[key].abs_vol_weight / (self.scms[key].specific_gravity *
specific_weight_water)
            temp_vol = temp_vol + self.scms[key].abs_vol_vol

        # at the end of this loop, the temp_vol variable contains the volume of all the SCMs
        # to this, we need to add the volume of air

        self.constituents["cement"].abs_vol_vol = self.constituents["cement"].abs_vol_weight /
(self.constituents["cement"].specific_gravity * specific_weight_water)
        self.constituents["water"].abs_vol_vol = self.constituents["water"].abs_vol_weight /
(self.constituents["water"].specific_gravity * specific_weight_water)

        # paste is all scms plus cement, air, and water
        self.abs_vol_paste = temp_vol + self.abs_vol_air + self.constituents["cement"].abs_vol_vol +
self.constituents["water"].abs_vol_vol

        # Finally, the volumes of aggregates are a function of the volume of the paste
        # Recall that we are assuming our absolute volume calculations have a volume of 27 cubic feet...

        self.constituents["coarse agg"].abs_vol_vol = (self.c_a_ratio_percent / 100.0) * (volume_conversion -
self.abs_vol_paste)
        self.constituents["coarse agg"].abs_vol_weight = (
            self.constituents["coarse agg"].abs_vol_vol * self.constituents["coarse agg"].specific_gravity *
specific_weight_water * (1.0 - self.model_1d.ar_ratio)
        )
        # NS - added accounting for recycled aggregates in the mix
        self.constituents["recycled agg"].abs_vol_weight = (
            self.constituents["coarse agg"].abs_vol_vol * self.constituents["recycled agg"].specific_gravity *
specific_weight_water * self.model_1d.ar_ratio
        )

        self.constituents["fine agg"].abs_vol_vol = (1.0 - self.c_a_ratio_percent / 100.0) * (volume_conversion -
self.abs_vol_paste)
        self.constituents["fine agg"].abs_vol_weight = (
            self.constituents["fine agg"].abs_vol_vol * self.constituents["fine agg"].specific_gravity *
specific_weight_water
        )

        self.concrete_weight = 0.0
        for key in self.constituents:
            self.concrete_weight = self.concrete_weight + self.constituents[key].abs_vol_weight

        for key in self.scms:
            self.concrete_weight = self.concrete_weight + self.scms[key].abs_vol_weight

        # NS - Adding workability as a part of the mix design. The workability is simply a calculated slump in cm


        lbcyd_to_kg_cm = 0.593276421

        cem_norm = ((self.cement_content * lbcyd_to_kg_cm) - 201.0) / (446.3 - 201.0)
        sand_norm = ((self.constituents["fine agg"].abs_vol_weight * lbcyd_to_kg_cm) - 384.0) / (827.0 - 384.0)
```

```
        coarse_norm = ((self.constituents["coarse agg"].abs_vol_weight + self.constituents["recycled
agg"].abs_vol_weight) * lbcyd_to_kg_cm - 1107.0) / (1218.0 - 1107.0)
        water_norm = ((self.constituents["water"].abs_vol_weight * lbcyd_to_kg_cm) - 164.0) / (186.0 - 164.0)

        self.workability = 36.22 - 12.47 * cem_norm - 27.03 * sand_norm - 7.39 * coarse_norm - 3.00 * water_norm

        # New compressive strength calculation self.
        self.compressive_strength = (13.352 * self.w_c_ratio ** (-1.081))
        if self.compressive_strength >= 2500:
            # NS - why is this here? It servers no purpose
            self.compressive_strength = self.compressive_strength_3day
        else:
            self.compressive_strength = (
            (51290.0) / (23.66 ** (self.w_c_ratio + 0.000378 * self.cement_content + 0.0279 *
self.air_content_percent))
            )

        self.strength_constr = 0


        # NS - included strength constrain to bound column solution field

        if self.compressive_strength <= 0.0:
            self.strength_constr = - 8000.0 / self.compressive_strength
        else:
            self.compressive_strength = self.compressive_strength

    def calc_cracking_time(self):
        # NS - The whole point of this function is to calculate the time till cracking for the 1D diffusion model
        # NS - all of the following parameters are taken as the upper bound or mean of their distribution if
designated
        ft = 544  # Tensile strength (psi) -mean
        E = 4350000  # Modulus of elasticity (psi) -mean
        tp = 0.0005  # Porous region thickness (ft) -mean
        alpha = 0.622  # Alpha -upper bound
        icorr = 0.0015  # Corrosion rate (A/ft^2) -mean
        phi = 2  # Creep coefficient of concrete
        v = 0.28  # Poisson's Ratio
        rho_r = 225  # Density of rust
        rho_s = 490  # Density of steel

        # This can be made iterative with the 1D model but wont change the time at all
        cover_depth = 2.0 / 12.0  # Assume rough cover depth in ft

        E_eff = E / (1 + phi)
        a = (self.model_1d.diam_rebar + 2*tp / 12)/2
        b = cover_depth + a
        tcrit = cover_depth * ft / E_eff * ((a ** 2 + b ** 2)/(b ** 2 - a ** 2) + v)
        # The 32.14 is the factor for gravitational acceleration in ft/sec^2
        Wporous = pi * rho_r * 32.14 * tp / 12 * self.model_1d.diam_rebar / 12
        Wexpand = pi * rho_r * 32.14 * tp / 12 * (self.model_1d.diam_rebar / 12 + 2 * tp / 12) * tcrit / 2

        Wcrit = rho_s / (rho_s - alpha * rho_r) * (Wporous + Wexpand)
        kp = (1 / alpha) * pi * self.model_1d.diam_rebar / 12 * icorr
        self.time_to_crack = int((Wcrit ** 2) / (2 * kp))

    # NS - added a function that can generate boundary conditions
    def calc_boundary_condition(self):
```

```
        # 5 different boundary condition cases
        # NS - Build in the boundary conditions for the 1D diffusion model
        # NS -  These boundary conditions are based on Life 365
        max_concentration = 0.0
        time_to_max = 200.0
        if self.site_location.exposure_class == 1.0:
            max_concentration = self.concrete_weight * 0.008
            time_to_max = 1.0
        elif self.site_location.exposure_class == 2.0:
            max_concentration = self.concrete_weight * 0.01
            time_to_max = 10.0
        elif self.site_location.exposure_class == 3.0:
            max_concentration = self.concrete_weight * 0.006
            time_to_max = 15.0
        elif self.site_location.exposure_class == 4.0:
            max_concentration = self.concrete_weight * 0.006
            time_to_max = 30.0
        elif self.site_location.exposure_class == 5.0:
            max_concentration = self.concrete_weight * 0.008
            time_to_max = 200.0

        # NS - build the boundary condition vector from the sloped and constant parts.
        if int(self.service_life_parameters.service_life - self.time_to_crack) <= time_to_max:
            self.boundary_condition = linspace(0, max_concentration, int(time_to_max * 12
/self.model_1d.delta_time))
        else:
            bc1 = linspace(0, max_concentration, int(time_to_max * 12 / self.model_1d.delta_time))
            bc2 = ones((int(self.service_life_parameters.service_life - time_to_max - self.time_to_crack) * 12 /
self.model_1d.delta_time)) * max_concentration
            self.boundary_condition = concatenate((bc1, bc2), axis=0)

    # NS -  Diffusion Coefficient calculation function
    def diffusion_coefficient(self):
            self.d_28 = 2.17e-12 * exp(self.w_c_ratio / 0.279)
            if self.scms["fly ash"].replacement_percentage > 0.0:
                self.m = 0.26 + 0.4*(self.scms["fly ash"].replacement_percentage / 50.0)
                self.d_ref = self.d_28 * (0.170 + 0.829 * exp(-1.0 * self.scms["fly ash"].replacement_percentage / 6.07))
            elif self.scms["silica fume"].replacement_percentage > 0.0:
                self.m = 0.26
                self.d_ref = self.d_28 * (0.206 + 0.794 * exp(-1.0 * self.scms["silica fume"].replacement_percentage /
2.51))
            elif self.scms["metakaolin"].replacement_percentage > 0.0:
                self.m = 0.26
                self.d_ref = self.d_28 * (0.191 + 0.809 * exp(-1.0 * self.scms["metakaolin"].replacement_percentage /
6.12))
            elif self.scms["slag"].replacement_percentage > 0.0:
                self.m = 0.26 + 0.4 * (self.scms["slag"].replacement_percentage / 70.0)
                self.d_ref = self.d_28
            # NS - allow for NO SCM replacement in the model
            else:
                self.m = 0.26
                self.d_ref = self.d_28

    # NS - Random recycled aggregate placement function. Could also be used to place any type of aggregate in the
analysis depth
    def calc_agg_placement(self):
        self.flag = zeros(shape=(self.model_1d.slices, 1))
        dx = self.model_1d.depth / float(self.model_1d.slices)
```

```
        # only run if there is a replacement ratio greater than 0 to prevent break
        if self.model_1d.ar_ratio > 0.0:
            num_agg = floor(self.model_1d.ar_ratio * self.model_1d.depth / self.model_1d.agg_size) # number of
recycled agg in analysis

            if num_agg > 0.0:
                x_agg = (self.model_1d.depth / num_agg)
                for i in range(0, int(num_agg)):
                    # start coordinate for aggregate placement
                    x_agg_s = int(floor(i * x_agg / dx) + ceil((ceil(100 * random.random()) / 100 * (x_agg -
self.model_1d.agg_size)) / dx))
                    # end coordinate for aggregate placement
                    x_agg_e = int(floor(x_agg_s + self.model_1d.agg_size / dx))
                    self.flag[x_agg_s:x_agg_e] = 1

    def calc_service_life(self):

        # NS - added line to run no diffusion model
        if self.service_life_parameters.service_life_model == 0.0:
            self.cover_depth = 0.06

        # NS - 1D diffusion model
        elif self.service_life_parameters.service_life_model == 1.0:
            # all the months and their duration (days)
            months = [31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31]
            temperature_new = zeros(shape=(1, len(months)/self.model_1d.delta_time))
            months_new = zeros(shape=(1, len(months) / self.model_1d.delta_time))

            for i in range(0, len(months) / self.model_1d.delta_time):
                temperature_new[0, i] = sum(self.site_location.temperature[i * self.model_1d.delta_time : i *
self.model_1d.delta_time + self.model_1d.delta_time]) / self.model_1d.delta_time
                months_new[0, i] = sum(months[i * self.model_1d.delta_time : i * self.model_1d.delta_time +
self.model_1d.delta_time])

            months = months_new
            temperature = temperature_new

            t = tile(months, int(self.service_life_parameters.service_life - self.time_to_crack))
            t = cumsum(t)
            T = tile(temperature, int(self.service_life_parameters.service_life - self.time_to_crack))

            # slice size in the analysis
            delta_x = self.model_1d.depth / float(self.model_1d.slices)

            # Coefficients for the temperature impact on diffusion coefficient
            U = 35000.0
            R = 8.314462
            T_ref = 21.0  # T_ref in degree C
            T_ref = T_ref + 273.15  # T_ref in degree K

            # NS - Preallocate all of the matrices used to do the 1D model
            r = zeros(shape=(len(t), self.model_1d.slices))
            D_c = zeros(shape=(len(t), self.model_1d.slices))  # Diffusion coefficient at time, slice
            A = zeros(shape=(self.model_1d.slices, self.model_1d.slices))
            B = zeros(shape=(self.model_1d.slices, self.model_1d.slices))
            c = zeros(shape=(self.model_1d.slices, 1))  # chloride level at slice

            # NS - Calculate the diffusion coefficient with respect to time and temperature and set inital
```

```
contamination
        for j in range(0, len(t)):
            for s in range(0, self.model_1d.slices):
                if self.flag[s] == 1:
                    if j == 0:
                        D_c[j, s] = self.model_1d.d_rma  # set slices that are recycled aggregates to have their
contamination and diffusion values
                        c[s] = self.model_1d.rec_agg_c
                    else:
                        D_c[j, s] = self.model_1d.d_rma * exp((U/R) * (1 / T_ref - 1 / T[0, j]))
                else:
                    if j == 0:
                        D_c[j, s] = self.d_28
                        c[s] = 0.0  # initial contamination is 0 (could be changed to account for recylced aggregates)
                    else:
                        D_c[j, s] = (self.d_ref * (28.0 / t[j]) ** self.m ) * exp((U/R) * (1 / T_ref - 1 / T[0, j]))

        # NS - Actually run the 1D diffusion model calculation
        for j in range(0, len(t)):
            for s in range(0, self.model_1d.slices):
                r[j, s] = D_c[j, s]*60.0*60.0*24.0*30*self.model_1d.delta_time / (2.0 * delta_x ** 2.0)
                # force first and las entry to be unity
                if s == 0:
                    A[s, s] = 1.0
                    B[s, s] = 1.0
                elif s == (self.model_1d.slices - 1):
                    A[s, s] = 1.0
                    B[s, s] = 1.0
                else:
                    A[s, s] = 1 + 2 * r[j, s]
                    A[s, s+1] = -r[j, s]
                    A[s, s-1] = -r[j, s]
                    B[s, s] = 1 - 2 * r[j, s]
                    B[s, s+1] = r[j, s]
                    B[s, s-1] = r[j, s]

            # NS - Apply the Boundary Conditions
            c[0] = self.boundary_condition[j]
            A_inv = linalg.inv(A)
            C = matmul(matmul(A_inv, B), c)
            c = C

            # NS - When the 1D simulation has reached the service life find the corresponding cover needed
            if j == len(t) - 1:
                temp = c - self.service_life_parameters.chloride_rebar_threshold
                # NS - add line to prevent the 1D model from breaking when contaminated agg prevent convergence
                if min(temp) > 0.0:
                    new_cover = self.model_1d.depth
                else:
                    # find the slice past the last slice that fails
                    m = max(i for i in temp if i < 0)
                    idx = where(temp == m)
                    find = idx[0]
                    new_cover = delta_x * find[0]

        self.cover_depth = new_cover

        # NS - Apply same cover depth constraints as below
```

```
            self.cover_constr = 0
            if self.cover_depth < self.service_life_parameters.cover_depth_lower:
                # print "Cover depth, %f, is too low!" % self.cover_depth
                self.cover_constr = self.service_life_parameters.cover_depth_lower - self.cover_depth

            if self.cover_depth > self.service_life_parameters.cover_depth_upper:
                # print "Cover depth, %f, is too high!" % self.cover_depth
                self.cover_constr = self.cover_depth - self.service_life_parameters.cover_depth_upper

        else:
            # A steady state version of the service life prediction

            # Note that service life has to be in seconds.  Therefore we make this temporary variable:
            year_into_seconds = 3.15569e7

            self.cover_depth = (
                2.0 * sqrt(self.d_ref * self.service_life_parameters.service_life * year_into_seconds) *
                special.erfinv(
                    1.0 - self.service_life_parameters.chloride_rebar_threshold /
self.service_life_parameters.chloride_exposure)
            )

            self.cover_constr = 0
            if self.cover_depth < self.service_life_parameters.cover_depth_lower:
                # print "Cover depth, %f, is too low!" % self.cover_depth
                self.cover_constr = self.service_life_parameters.cover_depth_lower - self.cover_depth

            if self.cover_depth > self.service_life_parameters.cover_depth_upper:
                # print "Cover depth, %f, is too high!" % self.cover_depth
                self.cover_constr = self.cover_depth - self.service_life_parameters.cover_depth_upper

    def calc_func_unit(self):
        if self.site_location.func_unit == 0.0:
            self.func_unit_vol = 1.0 * 1.0 * 1.0

        # NS - added column functional unit that must resist
        elif self.site_location.func_unit == 1.0:
            load = 1000  # Column load (kip)
            height = 12  # Column length (feet)
            area_str = (load * 1000.0) / self.compressive_strength  # Strength condition area required (in^2)

            # NS - Modulus of Elasticity calculated from AASHTO code 2007 (4th edition)
            Wc = 0.140 + self.compressive_strength / 1000.0  # Density of concrete (kip/ft^3)
            Emodulus = 33000.0 * Wc ** 1.5 * sqrt(self.compressive_strength / 1000.0)  # (ksi)

            # NS - Other inputs for the Euler buckling equation (stability strength area: area_stb)
            k = 1.0  # Assume effective length condition is fix-pin
            area_stb = sqrt((12.0 * load * (k * height * 12.0) ** 2.0) / (pi ** 2.0 * Emodulus))  # (in^2)

            self.column_constr = 0

            # NS - Check to see which criteria governs
            if area_stb > area_str:
                self.surface_area = 4 * height * 12.0 * sqrt(area_stb) * 0.00064516 # Area (m^2)
                self.func_unit_vol = height * 12.0 * area_stb * 0.00001639  # volume (m^3)
                if area_stb < 0:
                    self.column_constr = -1 + 256 / area_stb
            else:
```

```
            self.surface_area = 4 * height * 12.0 * sqrt(area_str) * 0.00064516 # Area (m^2)
            self.func_unit_vol = height * 12.0 * area_str * 0.00001639  # volume (m^3)
            if area_str < 0:
                self.column_constr = - 1 + 256 / area_str

        else:
            # The functional unit volume is 1 m, by 1 m, by the cover depth in m
            self.func_unit_vol = 1.0 * 1.0 * self.cover_depth

        # Now consider how to calculate the mass of each component, within the functional unit, in kilograms:

        # mass_func_unit / vol_func_unit = mass_abs_vol / vol_abs_vol
        # mass_func_unit = vol_func_unit * mass_abs_vol / vol_abs_vol

        # Units will get tricky here.  We actually had WEIGHT in the abs_vol calculation, not mass.  So we will
        # just convert
        #
        # (func_unit_vol m ^ 3) * (abs_vol_weight lb) * (1 / 1 yd ^3) * (kg / 2.20462262 lb) * (1.30795062 yd ^ 3 / m
^ 3)

        kg_into_lb = 2.20462262
        cy_into_cm = 1.30795062

        # In a set of loops, calculate mass of SCMs and constituents, and start running totals for
        # embodied energy and cost

        temp_mass = 0.0
        temp_mass_miles = 0.0
        temp_ee = 0.0
        # NS - added accounting for embodied carbon using the functional unit

        temp_ec = 0.0
        temp_cost = 0.0

        for key in self.scms:
            # self.scms[key].func_vol_mass = self.scms[key].abs_vol_weight * func_unit_vol * (1.0 / kg_into_lb) *
cy_into_cm
            self.scms[key].func_vol_mass = self.scms[key].abs_vol_weight * self.func_unit_vol * (1.0 / kg_into_lb)
            temp_mass = temp_mass + self.scms[key].func_vol_mass
            temp_mass_miles = temp_mass_miles + self.scms[key].func_vol_mass * self.scms[key].distance
            temp_ee = temp_ee + self.scms[key].embodied_energy * self.scms[key].func_vol_mass
            temp_ec = temp_ec + self.scms[key].embodied_carbon * self.scms[key].func_vol_mass
            temp_cost = temp_cost + self.scms[key].cost * self.scms[key].func_vol_mass

        for key in self.constituents:
            # NS - takes the weight and volume of the mix in lb and m^3 and converts them to kg and m^3
            self.constituents[key].func_vol_mass = self.constituents[key].abs_vol_weight * self.func_unit_vol * (1.0 /
kg_into_lb)
            temp_mass = temp_mass + self.constituents[key].func_vol_mass
            temp_mass_miles = temp_mass_miles + self.constituents[key].func_vol_mass *
self.constituents[key].distance
            temp_ee = temp_ee + self.constituents[key].embodied_energy * self.constituents[key].func_vol_mass
            temp_ec = temp_ec + self.constituents[key].embodied_carbon * self.constituents[key].func_vol_mass
            temp_cost = temp_cost + self.constituents[key].cost * self.constituents[key].func_vol_mass

        self.func_vol_total_mass = temp_mass
        self.func_vol_total_mass_miles = temp_mass_miles
```

```
        # Now that we have the mass miles we can use it to finish out the EE and cost calc...
        # high-range water-reducing mixture (HRWR) for low values of w/c ratio...
        # the calculation below is per volume unit (must multiply by v_t below)
        if self.w_c_ratio < 0.50:
            ee_hrwr = (-540.0 / 0.15)*self.w_c_ratio + 1800
            # NS - Adding embodied carbon impact for high-range water-reducing mixture here
            ec_hrwr = 1.0*self.w_c_ratio
        else:
            ee_hrwr = 0.0
            ec_hrwr = 0.0

        self.hrwr_energy = ee_hrwr * self.func_unit_vol
        self.hrwr_carbon = ec_hrwr * self.func_unit_vol

        temp_ee = (
            temp_ee + self.func_vol_total_mass_miles * self.components["transportation"].embodied_energy
            + self.hrwr_energy
        )
        temp_ec = (
            temp_ec + self.func_vol_total_mass_miles * self.components["transportation"].embodied_carbon
            + self.hrwr_carbon
        )
        temp_cost = temp_cost + self.func_vol_total_mass_miles * self.components["transportation"].cost

        self.func_vol_embodied_energy = temp_ee
        self.func_vol_embodied_carbon = temp_ec
        self.func_vol_total_cost = temp_cost

    # NS - added accounting for sequestered carbon
    def calc_sequestered_carbon(self):

        # only calculate sequesterd carbon if model is the column
        if self.carbon_sequestration.model_control == 1.0:
            # NS - Carbon sequestration potential
            cm = 0.0
            flag = 0
            for key in self.scms:
                if self.scms[key].replacement_percentage > 0.0:
                    cm = self.carbon_sequestration.alpha[self.carbon_sequestration.cement_type] - (self.scms[key].beta
* self.scms[key].replacement_percentage / 100.0)
                else:
                    flag = flag + 1

            if flag == 4:  # NS - if there are no SCMs then the cm is just alpha
                cm = self.carbon_sequestration.alpha[self.carbon_sequestration.cement_type]

            if cm < 0.0:  # NS - ensures that if a very high SCM replacement is used it wont make sequestered carbon
negative
                cm = 0.0

            # NS - Carbonation depth calculation
            k1 = self.carbon_sequestration.carbon_expo[self.carbon_sequestration.exposure].k1  # pulls proper k1
from dict
            n = self.carbon_sequestration.carbon_expo[self.carbon_sequestration.exposure].n  # pulls proper n from
dict
            if self.carbon_sequestration.cement_type == "Type I" or "Type II":
                R = 0.0016 * (self.compressive_strength * 0.00689476) ** 3.106  # NS - Strength must be in MPa
            else:
```

```
            R = 0.0018 * (self.compressive_strength * 0.00689476) ** 2.862

        x = sqrt((2.0 * self.carbon_sequestration.carbon_conc / 516000.0 *
self.service_life_parameters.service_life) / R) * \
            (sqrt(self.carbon_sequestration.k0 * k1 * self.carbon_sequestration.k2) * (1 /
self.service_life_parameters.service_life) ** n) \

        # NS - Carbonated Volume
        volume_c= self.surface_area * x
        if volume_c > self.func_unit_vol:
            volume_c = self.func_unit_vol

        # NS - Carbon Sequestered

        self.sequestered_carbon = self.carbon_sequestration.degree_carbonation * cm * volume_c *
self.cement_content * 0.593276421  # cement weight in kg/m^3

    else:  # If carbon sequestration is not considered...
        self.sequestered_carbon = 0.0

    self.func_vol_embodied_carbon = self.func_vol_embodied_carbon - self.sequestered_carbon
```

B.5 PROBLEM MODULES

The problem modules script is responsible for plotting output data and running the monte-carlo simulation to verify the solution space if requested

```
__author__ = 'nast1697'
# My Python files:
from wrappermodules import *
from concretemodules import *

# Other Python modules
import numpy as np
from pylab import *
from mpl_toolkits.mplot3d import Axes3D

# Parallel Plot modules
import subprocess
import pandas as pd
from pandas.tools.plotting import parallel_coordinates
import matplotlib.pyplot as plt
# import seaborn # NS - this can just make plots grey scale and higher contrast

def plotting_2D(resultFileName,objectives,decisions):
    # NS - This plotting code will currently plot the mixes using flyash, silicafume and metakaolin.
    # Read in the input files

    print "Plotting in 2D, nothing to see here"

    lines = [line.rstrip('\n') for line in open(resultFileName)]

    # Remove the first 12 lines and the last two, because they have no data
    # in them
    lines = lines[12:(len(lines) - 2)]
    fileData = []
    for line in lines:
        myrow = [float(value) for value in line.split()]
        fileData.append(myrow)

    num_rows = len(fileData)
    num_cols = len(decisions) + len(objectives)

    for key in decisions:
        decisions[key].data = np.empty(num_rows)

    for key in objectives:
        objectives[key].data = np.empty(num_rows)

    for row_it in range(len(fileData)):
        col_it = 0
        for key in decisions:
            # NS - used to debug the issues related to the hard coding of decisions in the mix_for_borg file
            # print "Decision %s is column %i" % (key, col_it)
            decisions[key].data[row_it] = fileData[row_it][col_it]
            col_it = col_it + 1
```

```
        for key in objectives:
            # NS - used to debug the issues related to the hard coding of objectives in the mix_for_borg file
            # print "Objective %s is column %i" % (key, col_it)
            if key == "strength":
                objectives[key].data[row_it] = (-1.0) * fileData[row_it][col_it]
            else:
                objectives[key].data[row_it] = fileData[row_it][col_it]
            col_it = col_it + 1

    x = objectives["cost"].data
    y = objectives["ee"].data
    s = (100.0)*np.ones(num_rows)
    # This is the variable for color which corresponds with strength
    c = objectives["strength"].data
    constituents = [round(val) for val in decisions["scm"].data]
    # find the fly ash and silica fume values
    fly_ash_indices = [i for i in range(num_rows) if constituents[i] == 1.0]
    silica_fume_indices = [i for i in range(num_rows) if constituents[i] == 2.0]
    metakoalin_indices = [i for i in range(num_rows) if constituents[i] == 3.0]
    slag_indices = [i for i in range(num_rows) if constituents[i] == 4.0]
    figure(resultFileName)
    # NS - add if statements so colorbar isnt messed up due to no data of a certain type being in the plot
    if len(fly_ash_indices) > 0:
        scatter(x[fly_ash_indices], y[fly_ash_indices], s[fly_ash_indices], c[fly_ash_indices], marker='^', alpha=0.75)
    if len(silica_fume_indices) > 0:
        scatter(x[silica_fume_indices], y[silica_fume_indices], s[silica_fume_indices], c[silica_fume_indices],
marker='o',
         alpha=0.75)

    # NS - Added a line to plot metakaolin as well
    if len(metakoalin_indices) > 0:
        scatter(x[metakoalin_indices], y[metakoalin_indices], s[metakoalin_indices], c[metakoalin_indices],
marker='s',
         alpha=0.75)

    # NS - Added a line to plot slag as well
    if len(slag_indices) > 0:
        scatter(x[slag_indices], y[slag_indices], s[slag_indices], c[slag_indices], marker='8',
         alpha=0.75)

    title(resultFileName)
    xlabel('Cost [$]')
    ylabel('EE [MJ]')
    colorbar().set_label('Strength [psi]')

    show()
    pass


def plotting_3D(resultFileName,objectives,decisions):
    # NS - This plotting code will currently plot the mixes using flyash, silicafume and metakaolin.
    # Read in the input files

    print "Plotting in 3D, nothing to see here"
    fig = plt.figure(resultFileName)
    ax = fig.add_subplot(111, projection='3d')
    lines = [line.rstrip('\n') for line in open(resultFileName)]
```

```
    # Remove the first 12 lines and the last two, because they have no data
    # in them
    lines = lines[12:(len(lines) - 2)]
    fileData = []
    for line in lines:
        myrow = [float(value) for value in line.split()]
        fileData.append(myrow)

    num_rows = len(fileData)
    num_cols = len(decisions) + len(objectives)

    for key in decisions:
        decisions[key].data = np.empty(num_rows)

    for key in objectives:
        objectives[key].data = np.empty(num_rows)

    for row_it in range(len(fileData)):
        col_it = 0
        for key in decisions:
            # NS - used to debug the issues related to the hard coding of decisions in the mix_for_borg file
            # print "Decision %s is column %i" % (key, col_it)
            decisions[key].data[row_it] = fileData[row_it][col_it]
            col_it = col_it + 1

        for key in objectives:
            # NS - used to debug the issues related to the hard coding of objectives in the mix_for_borg file
            # print "Objective %s is column %i" % (key, col_it)
            if key == "strength":
                objectives[key].data[row_it] = (-1.0) * fileData[row_it][col_it]
            else:
                objectives[key].data[row_it] = fileData[row_it][col_it]
            col_it = col_it + 1

    x = objectives["cost"].data
    y = objectives["ee"].data
    z = objectives["ec"].data
    # Choose which direction to make the z axis
    zdir = "z"
    s = (100.0)*np.ones(num_rows)
    # This is the variable for color which corresponds with strength
    c = objectives["strength"].data
    constituents = [round(val) for val in decisions["scm"].data]

    # find the fly ash and silica fume values
    fly_ash_indices = [i for i in range(num_rows) if constituents[i] == 1.0]
    silica_fume_indices = [i for i in range(num_rows) if constituents[i] == 2.0]
    metakoalin_indices = [i for i in range(num_rows) if constituents[i] == 3.0]
    slag_indices = [i for i in range(num_rows) if constituents[i] == 4.0]
    if len(fly_ash_indices) > 0:
        p = ax.scatter(x[fly_ash_indices], y[fly_ash_indices], z[fly_ash_indices], zdir, s[fly_ash_indices],
c[fly_ash_indices], marker='^')
    if len(silica_fume_indices) > 0:
        p = ax.scatter(x[silica_fume_indices], y[silica_fume_indices], z[silica_fume_indices], zdir,
s[silica_fume_indices], c[silica_fume_indices], marker='o',
         alpha=0.75)
    if len(metakoalin_indices) > 0:
```

```
        p = ax.scatter(x[metakoalin_indices], y[metakoalin_indices], z[metakoalin_indices], zdir,
s[metakoalin_indices], c[metakoalin_indices], marker='s',
         alpha=0.75)
    if len(slag_indices) > 0:
        p = ax.scatter(x[slag_indices], y[slag_indices], z[slag_indices], zdir, s[slag_indices], c[slag_indices],
marker='*',
         alpha=0.75)

    # ax.set_title(resultFileName)
    ax.set_xlabel('Cost $')
    ax.set_ylabel('EE [MJ]')
    ax.set_zlabel('EC [kg CO2]')

    plt.colorbar(p).set_label('Strength [psi]')
    plt.show()
    plt.savefig('3D_plot.jpg')
    pass


def parallel_plot_objectives(resultFileName, objectives, decisions):

    print "Parallel Plotting in 3D, nothing to see here"
    # NS - create new file that is only the data no additional information
    plotdataName = "NoHeaders.txt"
    file = open(plotdataName,"w")
    lines = [line.rstrip('\n') for line in open(resultFileName)]

    # Remove the first 12 lines and the last two, because they have no data
    # in them
    lines = lines[12:(len(lines) - 2)]

    file.write("\n".join(lines))
    file.close()

    # NS - inputs related to the parallel command construction function
    plotCodeName = "parallel.py"
    plotFileName = "parallelplot1"
    # columns is written so only the columns with objective data will be pulled
    col = (len(decisions), len(objectives) + len(decisions) - 1)
    # col = (0, len(objectives) + len(decisions) - 1)
    numcol = len(objectives)
    # numcol = len(objectives) + len(decisions)
    # currently 6 columns of data are being called thus 6 precisions must be provided
    precision = (0.1, 0.1, 0.1, 0.1, 0.1, 0.1)
    # precision = (0.1, 0.1, 0.1, 0.1, 0.1, 0.1,0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1)
    # names of the axes used on the plot
    names = ("Cost", "EE", "Strength", "Cover", "EC", "Slump")
    # names = ("Cement", "w/c", "air", "CAR", "SCM", "Replace", "RAC", "Cost", "EE", "Strength", "Cover", "EC",
"Slump")
    dataname = ("data")
    # four variable color code
    # color = (0.25, 0.5, 0.8, 0.5) # Light Blue
    # color = (0.25, 0.4, 0.8, 0.9) # Deep Blue
    color = (0.0, 0.28, 0.61, 0.2)  # Greyish blue
    # color = (0, 0, 1, 0.38)
    minima = (50, 1100, -12000, 0.0, 200, -22) # again there must be as many minima as columns
    # minima = (0, 40, -12000, 0.0, 50, -22)
    maxima = (100, 2200, -1000, 0.5, 300, -1)
```

```
    # maxima = (15, 130, -1000, 0.2, 90, -1)
    linewidth = (0.7, 0.7)

    parallelCommand = construct_parallel_command(plotCodeName, plotFileName, plotdataName, col, numcol,
precision, names, dataname, color, minima, maxima, linewidth)

    subprocess.call(parallelCommand)

    # NS - Alternative parallel plotting that requires some data output data file manipulation to work well.
    # data = pd.read_csv(resultFileName)

    # parallel_coordinates(fileData, 'Name', color=['#225ea8', '#7acde9', '#9afff7', '#5a668a', '#3c445c', '#cbbdc0',
'#38654b', '#415a44', '#a3e3aa', '#f3403f', '#f6706f', '#ffe0e1', '#ffe0c0' ], linewidth=5, alpha=.8)
    # plt.ylabel('Direction of Preference $\ \rightarrow$', fontsize=12)
    #
    # plt.savefig('parallel_plot.jpg')

def parallel_plot_all(resultFileName, objectives, decisions):

    print "Parallel Plotting in 3D, nothing to see here"
    # NS - create new file that is only the data no additional information
    plotdataName = "NoHeaders.txt"
    file = open(plotdataName,"w")
    lines = [line.rstrip('\n') for line in open(resultFileName)]

    # Remove the first 12 lines and the last two, because they have no data
    # in them
    lines = lines[12:(len(lines) - 2)]

    file.write("\n".join(lines))
    file.close()

    # NS - inputs related to the parallel command construction function
    plotCodeName = "parallel.py"
    plotFileName = "parallelplot1"
    # columns is written so only the columns with objective data will be pulled
    col = (0, len(objectives) + len(decisions) - 1)
    numcol = len(objectives) + len(decisions)
    # currently 6 columns of data are being called thus 6 precisions must be provided
    precision = (0.1, 0.1, 0.1, 0.1, 0.1, 0.1,0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1)
    # names of the axes used on the plot
    names = ("Cement", "w/c", "air", "CAR", "SCM", "Replace", "RAC", "Cost", "EE", "Strength", "Cover", "EC",
"Slump")
    dataname = ("data")
    # four variable color code
    # color = (0.25, 0.5, 0.8, 0.5) # Light Blue
    # color = (0.25, 0.4, 0.8, 0.9) # Deep Blue
    color = (0.0, 0.28, 0.61, 0.2)  # Greyish blue
    # color = (0, 0, 1, 0.38)
    # minima = (550, 0.25, 1, 55, 0, 0, 0, 65, 1600, -12000, 0.0, 260, -22)
    minima = (550, 0.25, 1, 55, 0, 0, 0, 0, 25, -12000, 0.0, 0, -22)
    # maxima = (700, 0.75, 6, 75, 5, 30, 1, 120, 3000, -1000, 0.5, 420, -1)
    maxima = (700, 0.75, 6, 75, 5, 30, 1, 5, 80, -1000, 0.2, 15, -1)
    linewidth = (0.7, 0.7)

    parallelCommand = construct_parallel_command(plotCodeName, plotFileName, plotdataName, col, numcol,
precision, names, dataname, color, minima, maxima, linewidth)
```

```
    subprocess.call(parallelCommand)


def parallel_plot_decisions(resultFileName, objectives, decisions):

    print "Parallel Plotting in 3D, nothing to see here"
    # NS - create new file that is only the data no additional information
    plotdataName = "NoHeaders.txt"
    file = open(plotdataName,"w")
    lines = [line.rstrip('\n') for line in open(resultFileName)]

    # Remove the first 12 lines and the last two, because they have no data
    # in them
    lines = lines[12:(len(lines) - 2)]

    file.write("\n".join(lines))
    file.close()

    # NS - inputs related to the parallel command construction function
    plotCodeName = "parallel.py"
    plotFileName = "parallelplot1"
    # columns is written so only the columns with objective data will be pulled
    col = (0, len(decisions) - 1)
    numcol = len(decisions)
    # currently 6 columns of data are being called thus 6 precisions must be provided
    precision = (0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1)
    # names of the axes used on the plot
    names = ("Cement", "w/c", "air", "CAR", "SCM", "Replace", "RAC")
    dataname = ("data")
    # four variable color code
    # color = (0.25, 0.5, 0.8, 0.5) # Light Blue
    # color = (0.25, 0.4, 0.8, 0.9) # Deep Blue
    color = (0.0, 0.28, 0.61, 0.2)  # Greyish blue
    minima = (550, 0.25, 1, 55, 0, 0, 0)
    maxima = (700, 0.75, 6, 75, 5, 30, 1)
    linewidth = (0.7, 0.7)

    parallelCommand = construct_parallel_command(plotCodeName, plotFileName, plotdataName, col, numcol,
precision, names, dataname, color, minima, maxima, linewidth)

    subprocess.call(parallelCommand)

    # NS - Alternative parallel plotting that requires some data output data file manipulation to work well.
    # data = pd.read_csv(resultFileName)

    # parallel_coordinates(fileData, 'Name', color=['#225ea8', '#7acde9', '#9afff7', '#5a668a', '#3c445c', '#cbbdc0',
'#38654b', '#415a44', '#a3e3aa', '#f3403f', '#f6706f', '#ffe0e1', '#ffe0c0' ], linewidth=5, alpha=.8)
    # plt.ylabel('Direction of Preference $\ \rightarrow$', fontsize=12)
    #
    # plt.savefig('parallel_plot.jpg')

def monte_carlo_sim(runMonteCarlo, numSimulations, fileName, serviceLife, chlorideExposure, fileType,
                    num_objectives,decisions,num_decisions,num_constraints):

    # NS - This needs to be removed form here and placed in its own script (not concretemodules.py or
wrappermodules.py)
    # runMonteCarlo = 0
```

```
if runMonteCarlo:

    print "Beginning Monte Carlo run with %d simulations" % numSimulations

    monte_carlo_objconstrs = np.empty([numSimulations, num_objectives + num_constraints])
    monte_carlo_decisions = np.empty([numSimulations, num_decisions])

    # list of lower and upper bounds of decisions.  The variables are:
    # cement, wc, air, ca, slag, scm, replace

    #lowers = [550.0, 0.25, 1.0, 55.0, 0.0, 0.5, 0.0]
    #uppers = [700.0, 0.75, 3.0, 75.0, 50.0, 3.5, 30.0]

    # Latin Hypercube Sampling adapted from: https://mathieu.fenniak.net/latin-hypercube-sampling/
    segmentSize = 1.0/float(numSimulations)

    dec_it = 0
    for key in decisions:

        # Each variable must have its own latin hypercube sample.  Since we are creating
        # an ensemble of samples, we need to make sure that each iteration gets a different
        # segment position for each column of the ensemble.  So, instead of iterating over the range
        # of segments in order, we permute the segments for each variable.  Of course, each variable
        # also gets a different plotting position WITHIN that segment too.

        thisVariablePermutation = np.random.permutation(numSimulations)

        # print("Variable %d has permutation starting with %.6f" % (dec_it, thisVariablePermutation[0]))

        for sim_it in range(numSimulations):
            mySegment = thisVariablePermutation[sim_it]
            segmentMin = (float(mySegment)+1.0)*segmentSize
            point = segmentMin + (random.random()*segmentSize)
            pointValue = point * (decisions[key].upper - decisions[key].lower) + decisions[key].lower
            if key == "scm":
                # See what the rounding is going to do
                test_round = round(pointValue)

                # This is a hack to make sure you don't get weird constituents since perhaps python is
                # rounding stuff up when it's supposed to round it down, etc.
                if test_round == 4.0:
                    test_round = 3.0
                elif test_round == 0.0:
                    test_round = 1.0

                monte_carlo_decisions[sim_it][dec_it] = test_round

            else:
                monte_carlo_decisions[sim_it][dec_it] = pointValue
        dec_it = dec_it + 1

    # Here are some important parameters for this particular Monte Carlo sampling.  You could technically
    # run multiple Monte Carlo simulations by replicating this whole code block multiple times.

    monte_carlo_filename = result_file_name(fileName, serviceLife, chlorideExposure, fileType)
```

```
for i in range(numSimulations):

    # Evaluate the problem

    # Decision variables:
    # cement content, w_c_ratio, air_content, c_a_ratio_percent, r_slag, r_choice, r_scm_replace

    # The order MUST MATCH the order of the OrderedDict() variable, decisions, above

    cement_content = monte_carlo_decisions[i][0]
    w_c_ratio = monte_carlo_decisions[i][1]
    air_content = monte_carlo_decisions[i][2]
    c_a_ratio_percent = monte_carlo_decisions[i][3]
    r_slag = monte_carlo_decisions[i][4]
    r_scm_choice = monte_carlo_decisions[i][5]
    r_scm_replace = monte_carlo_decisions[i][6]

    if r_scm_choice == 1:
        r_fly_ash = r_scm_replace
        r_silica_fume = 0.0
        r_metakaolin = 0.0
    elif r_scm_choice == 2:
        r_fly_ash = 0.0
        r_silica_fume = r_scm_replace
        r_metakaolin = 0.0
    elif r_scm_choice == 3:
        r_fly_ash = 0.0
        r_silica_fume = 0.0
        r_metakaolin = r_scm_replace
    #else:
        #print "Error, the SCM choice number is wrong!"

    # There is another way to do this, of course.  Ideally, we would mimic exactly the script that is
    # used to run Borg.  That way, someone could change only one file and it would be seamless.
    # Instead, we are mimicking what is going on in mix_for_borg_B1.py, but the main issue is that
    # there could be parameters that are different between them.  The solution is to create another
    # subprocess and communicate with it interactively (subprocess.communicate())

    # Anyway:

    # Arguments: Name, cost, embodied energy, embodied carbon
    components = dict()
    components["transportation"] = Component("transportation", 0.005, 0.02, 0.1)

    # Arguments: Name, cost, embodied energy, embodied carbon, distance, specific gravity
    constituents = dict()
    constituents["coarse agg"] = Constituent("coarse aggregate", 0.012, 0.10, 0.0053,50.0, 2.5)
    constituents["fine agg"] = Constituent("fine aggregate", 0.02, 0.08, 0.0051, 75.0, 2.63)
    constituents["cement"] = Constituent("cement", 0.0985, 5.9, 0.852, 20.0, 3.15)
    constituents["water"] = Constituent("water", 0.005, 0.01, 0.001, 20.0, 1.0)

    # Arguments: name, cost, embodied energy, embodided carbon, distance, specific gravity, replacement
percentage
    scms = dict()
    scms["slag"] = SCM("slag", 0.017, 1.6, 0.083, 500.0, 2.9, r_slag)
    scms["fly ash"] = SCM("fly ash", 0.03, 0.1, 0.008, 500.0, 2.4, r_fly_ash)
    scms["silica fume"] = SCM("silica fume", 0.18, 0.04, 0.15, 300.0, 2.25, r_silica_fume)
    scms["metakaolin"] = SCM("metakaolin", 0.045, 2.08, 0.45, 800.0, 2.5, r_metakaolin)
```

```
        # Arguments: service life, chloride exposure, chloride rebar threshold, cover depth lower, cover depth
upper
        serv_param = ServiceLifeParameters(serviceLife, chlorideExposure, 0.7, 0.02, 0.2)

        # Arguments: cement content, w/c ratio, air content percentage, c/a ratio percentage,
        # python objects: components, constituents, scms, serv_param

        mymix = MixDesign(cement_content, w_c_ratio, air_content, c_a_ratio_percent, components,
constituents, scms, serv_param)
        mymix.calc_absolute_volume()
        mymix.calc_service_life()
        mymix.calc_func_unit()

        monte_carlo_objconstrs[i][0] = mymix.func_vol_total_cost
        monte_carlo_objconstrs[i][1] = mymix.func_vol_embodied_energy
        monte_carlo_objconstrs[i][2] = mymix.compressive_strength
        monte_carlo_objconstrs[i][3] = mymix.cover_depth

    print "The Monte Carlo simulation is over, generating .csv file"

    nameHandle = open(monte_carlo_filename, 'w')

    # Print the header line
    nameHandle.write('cement, wc, air, ca, slag, scm, replace, cost, ee, compressive, cover \n')

    for i in range(numSimulations):
      for j in range(num_decisions):
        nameHandle.write('%.6f' % monte_carlo_decisions[i][j] + ',')
      for j in range(num_objectives):
        nameHandle.write('%.6f' % monte_carlo_objconstrs[i][j])
        if j == 3:
          nameHandle.write(',,\n')
          #nameHandle.flush()
        else:
          nameHandle.write(',')

    nameHandle.close()
  else:
    print "You chose to not run the Monte Carlo hence it is not running"
```