The Automated Modeling and Optimization of Part DNA Substructures Employing Evolutionary Computation

Daniel Tauritz, Bailey Eversmeyer

Missouri University of Science and Technology
Department of Computer Science

tauritzd@mst.edu, rbe62d@mst.edu

May 9, 2019

Overview

Part-DNA Overview

Goals:

- Model and map the flow of goods and components through a system

Part-DNA Overview

Goals:

- Model and map the flow of goods and components through a system
- Track the changes to components over time

Part-DNA Overview

Goals:

- Model and map the flow of goods and components through a system
- Track the changes to components over time
- Help identify relationships between components

Part-DNA Overview

Goals:

- Model and map the flow of goods and components through a system
- Track the changes to components over time
- Help identify relationships between components
- Makes analyzing the system easier

How We Fit into the Part-DNA Model

1. Choose a substructure of the Part-DNA Model

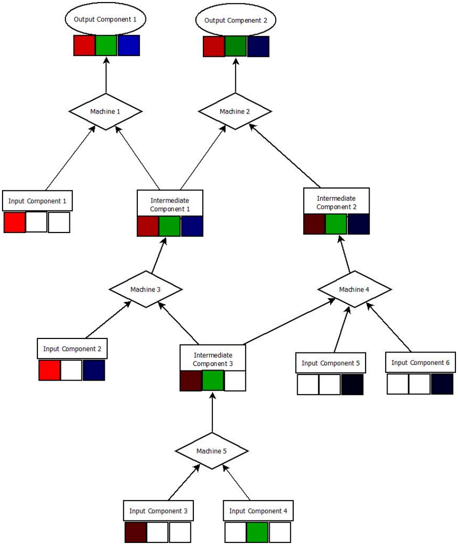
How We Fit into the Part-DNA Model

1. Choose a substructure of the Part-DNA Model
2. Modeling the substructure (GP)

Modeling the Substructure

1. Map the layout into a well-defined ordering

Our Model Concept



Modeling the Substructure

1. Map the layout into a well-defined ordering
2. Gather data on input-output component transformations

Modeling the Substructure

1. Map the layout into a well-defined ordering
2. Gather data on input-output component transformations
3. Model the transformations of components

How We Fit into the Part-DNA Model

1. Choose a substructure of the Part-DNA Model
2. Modeling the substructure (GP)

How We Fit into the Part-DNA Model

1. Choose a substructure of the Part-DNA Model
2. Modeling the substructure (GP)
3. Optimize input combinations (MOEA)

Optimizing the Substructure

With the model in hand:

1. Gather data on possible input components

Optimizing the Substructure

With the model in hand:

1. Gather data on possible input components
2. Test new input combinations to map Pareto Trade-Off surface

Evolutionary Algorithms (EAs)

- EAs are stochastic, population-based local search algorithms inspired by neo-Darwinian Evolution Theory

Evolutionary Algorithms (EAs)

- EAs are stochastic, population-based local search algorithms inspired by neo-Darwinian Evolution Theory
- Work by generating solutions, and testing fitness

Evolutionary Algorithms (EAs)

- EAs are stochastic, population-based local search algorithms inspired by neo-Darwinian Evolution Theory
- Work by generating solutions, and testing fitness
- Explore search space through recombination and mutation

Evolutionary Algorithms (EAs)

- EAs are stochastic, population-based local search algorithms inspired by neo-Darwinian Evolution Theory
- Work by generating solutions, and testing fitness
- Explore search space through recombination and mutation
- Best population members chosen via Survival-of-the-fittest

Evolutionary Algorithms (EAs)

- EAs are stochastic, population-based local search algorithms inspired by neo-Darwinian Evolution Theory
- Work by generating solutions, and testing fitness
- Explore search space through recombination and mutation
- Best population members chosen via Survival-of-the-fittest
- Individual **A** is better than individual **B** if **A** has a higher fitness than **B**

Genetic Programming (GP)

- Variable-size hierarchical representation vs fixed-size linear for EAs

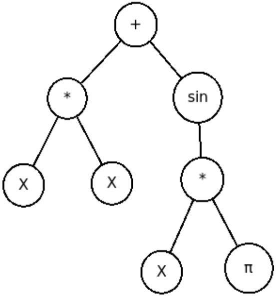
Genetic Programming (GP)

- Variable-size hierarchical representation vs fixed-size linear for EAs
- Generally a tree-like structure that can model functions:

Genetic Programming (GP)

- Variable-size hierarchical representation vs fixed-size linear for EAs
- Generally a tree-like structure that can model functions:

$$Y = X^2 + sin(X * \pi) \tag{1}$$



Multi-Objective EAs (MOEAs)

- Extension of regular EAs which maps multiple objective values to single fitness

Multi-Objective EAs (MOEAs)

- Extension of regular EAs which maps multiple objective values to single fitness
- Objectives usually conflict

Multi-Objective EAs (MOEAs)

- Extension of regular EAs which maps multiple objective values to single fitness
- Objectives usually conflict
- Uses *Dominance* in place of fitness

Multi-Objective EAs (MOEAs)

- Extension of regular EAs which maps multiple objective values to single fitness
- Objectives usually conflict
- Uses *Dominance* in place of fitness
- Individual **A** dominates individual **B** iff:

Multi-Objective EAs (MOEAs)

- Extension of regular EAs which maps multiple objective values to single fitness
- Objectives usually conflict
- Uses *Dominance* in place of fitness
- Individual **A** dominates individual **B** iff:
    - **A** is no worse than **B** in all objectives

Multi-Objective EAs (MOEAs)

- Extension of regular EAs which maps multiple objective values to single fitness
- Objectives usually conflict
- Uses *Dominance* in place of fitness
- Individual **A** dominates individual **B** iff:
  - **A** is no worse than **B** in all objectives
  - **A** is strictly better than **B** in at least one objective

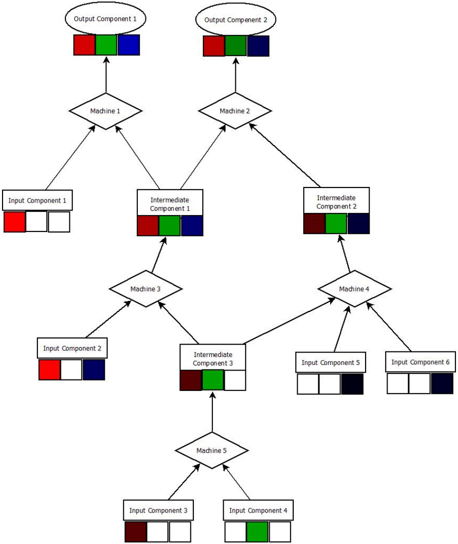
MOEA Dominance

- **A** *dominates* **B**
  - **A:** `Accuracy 60%, Affordability 2`
  - **B:** `Accuracy 50%, Affordability 2`

MOEA Dominance

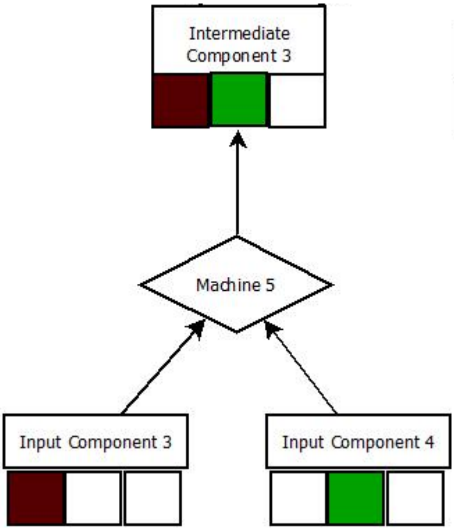
- **A** *dominates* **B**
  - **A:** `Accuracy 60%, Affordability 2`
  - **B:** `Accuracy 50%, Affordability 2`
- **A** does *not dominate* **B**
  - **A:** `Accuracy 60%, Affordability 1`
  - **B:** `Accuracy 50%, Affordability 2`

GP Section



Output Component 1
Output Component 2
Machine 1
Machine 2
Input Component 1
Intermediate Component 1
Intermediate Component 2
Machine 3
Machine 4
Input Component 2
Intermediate Component 3
Input Component 5
Input Component 6
Machine 5
Input Component 3
Input Component 4

GP Section


Intermediate Component 3
Machine 5
Input Component 3
Input Component 4

GP Process

Given a dataset of input-output combinations
For each output attribute:

- Generate population of randomized functions from the input domain

GP Process

Given a dataset of input-output combinations
For each output attribute:

- Generate population of randomized functions from the input domain
- Assign fitness value based on error across the dataset

GP Process

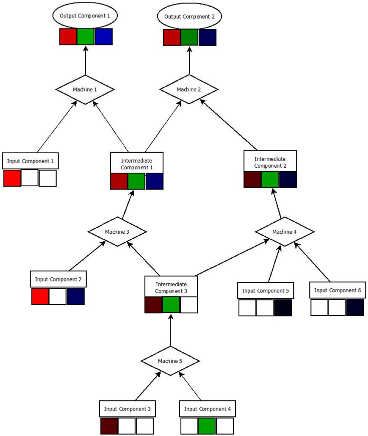
Given a dataset of input-output combinations
For each output attribute:

- Generate population of randomized functions from the input domain
- Assign fitness value based on error across the dataset
- Explore the function domain through recombination and mutation of functions

Repeat for each transformation object

MOEA Section

With the modeled functions in hand, we apply our MOEA to the whole process to optimize for the output parameters



MOEA Process

Given a dataset of possible inputs and desired outputs:

- Generate population of randomly chosen inputs

MOEA Process

Given a dataset of possible inputs and desired outputs:

- Generate population of randomly chosen inputs
- Simulate the system with each input combination

MOEA Process

Given a dataset of possible inputs and desired outputs:

- Generate population of randomly chosen inputs
- Simulate the system with each input combination
- Assign fitness values for Accuracy and Affordability

MOEA Process

Given a dataset of possible inputs and desired outputs:

- Generate population of randomly chosen inputs
- Simulate the system with each input combination
- Assign fitness values for Accuracy and Affordability
- Rate solutions based on their Pareto score

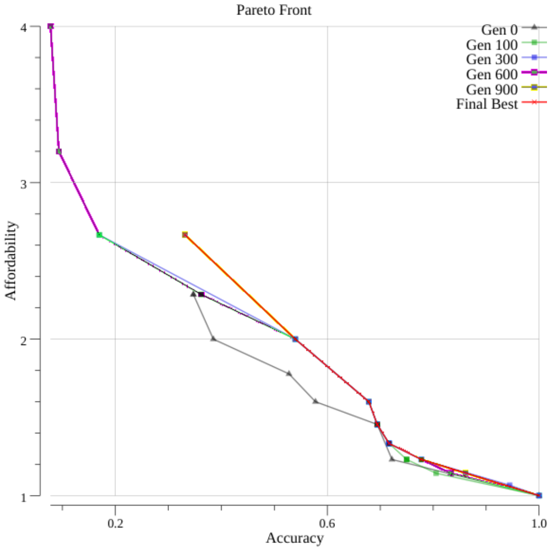
MOEA Process

Given a dataset of possible inputs and desired outputs:

- Generate population of randomly chosen inputs
- Simulate the system with each input combination
- Assign fitness values for Accuracy and Affordability
- Rate solutions based on their Pareto score
- Explore the input combination domain through recombination and mutation of solutions

End with a selection of Pareto Optimal solutions, and associated trade-off information.

Example Pareto Front over Time



Future of the Project

- Realistic datasets, both transformation machines and full substructure simulation

Future of the Project

- Realistic datasets, both transformation machines and full substructure simulation
- Possibility for optimizing full substructure layout/ordering

Questions?

References

- Dr. Tauritz's Intro to EA class slides
  - `http://web.mst.edu/~tauritzd/courses/ec/`