



UNIVERSITY  
OF APPLIED SCIENCES  
UPPER AUSTRIA

Oliver Krauss

# Exploring the Use of Natural Language Processing Techniques for Enhancing Genetic Improvement

Melbourne (online) - 20. May 2023

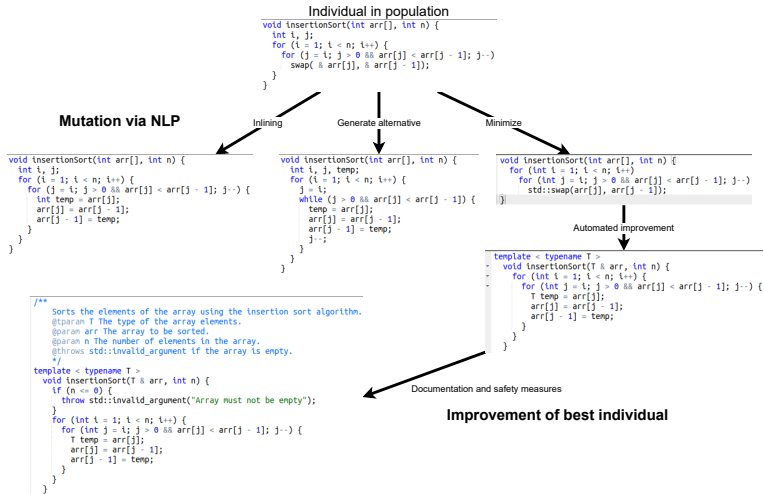
12th International Workshop on Genetic Improvement @ICSE 2023

# Didn't we just have this presentation?

## Towards Objective-Tailored Genetic Improvement Through Large Language Models - Sungming Kang, Shin Yoo

- Similar ideas
- Large Language Models (LLMs) are now used for everything
- Should GI jump on the bandwagon?
  - At least two independent groups think so

# Core concepts



# Uses of LLMs in GI - Mutation

- Recursion vs. iteration
- Function inlining
- API replacements
- Repair of GI generated mutations
- Non-functional considerations
- ...

# Mutation Example I

Listing 1: int array to Integer array. (Listing 6 in [1])

```
public class C315640{
    private static Integer [] toIntegerArray(int []
        array){
        Integer [] finalArray = new
            Integer[array.length];
        for (int i=0; i<array.length; i++){
            finalArray[i] = array[i];
            finalArray[i] = array[i];
        }
        return finalArray;
    }
}
```

# Mutation Example II

## Listing 2: Repair

```
public class C315640{
    private static Integer[] toIntegerArray(int []
        array){
        Integer[] finalArray = new
            Integer[array.length];
        for (int i=0; i<array.length; i++){
            finalArray[i] = array[i];
        }
        return finalArray;
    }
}
```

# Mutation Example III

## Listing 3: Simplification

```
public class C315640 {  
    private static Integer[] toIntegerArray(int []  
        array) {  
        return Arrays.stream(array).boxed()  
            .toArray(Integer []::new);  
    }  
}
```

# Mutation Example IV

## Listing 4: Security

```
public class C315640 {  
    private static Integer[] toIntegerArray(int []  
        array) {  
4      Objects.requireNonNull(array ,  
4          "Input array must not be null.");  
        return Arrays.stream(array).boxed()  
            .toArray(Integer []::new);  
    }  
}
```



# Uses of LLMs in GI - Explanation

- Patch explanation
- Adding documentation
- Type hints (in python)
- ...

# Explanation Example I

Listing 5: int array to Integer array. (Listing 6 in [1])

```
public class C315640{
    private static Integer [] toIntegerArray(int []
        array){
        Integer [] finalArray = new
            Integer[array.length];
        for (int i=0; i<array.length; i++){
            finalArray[i] = array[i];
            finalArray[i] = array[i];
        }
        return finalArray;
    }
}
```

# Explanation Example II

## Explanation

Line 5 assigns the value of the int at the current index of the input array (`array[i]`) to the corresponding index of the finalArray (`finalArray[i]`). It is duplicated in the code which will have no effect on the result.

# Explanation Example III

## Listing 6: Added Documentation

```
/**
 * This class provides a utility method to convert an int array to an Integer
 * array.
 */
public class C315640 {
    /**
     * Converts an array of primitive integers to an array of Integer objects.
     *
     * @param array the int array to be converted
     * @return an Integer array containing the elements from the input array
     * @throws NullPointerException if the input array is null
     */
    private static Integer[] toIntegerArray(int[] array) {
        // Null check for the input array
        Objects.requireNonNull(array, "Input array must not be null.");
        // Convert the int array to Integer array
        return Arrays.stream(array)
            .boxed().toArray(Integer[]::new);
    }
}
```

# Challenges

- Cost of LLM APIs / hosting
  - FlanT5-xl: 68GB of data
- Supported Languages
  - Available for Java, C, Python
  - Other languages?
  - StarCoder supports 80+ Languages  
<https://arxiv.org/pdf/2305.06161.pdf>

# Outlook

- Integration in open source framework `amaru.dev`
- Prompt-evaluation with different LLMs
- Using GI to improve LLM produced code (off by one errors)

# Contact

Code available under the MIT License at <https://amaru.dev>



**Oliver Krauss**

[oliver.krauss@fh-  
hagenberg.at](mailto:oliver.krauss@fh-hagenberg.at)

+43 (0)50804-27195

# Bibliography I

- [1] S. A. Licorish and M. Wagner, “Dissecting copy/delete/replace/swap mutations: Insights from a gin case study”, in *GECCO 2022*, Boston, Massachusetts: Association for Computing Machinery, 2022, pp. 1940–1945. [Online]. Available: <https://doi.org/10.1145/3520304.3533970>.