

**Resumo do artigo Proposta de dissertação:
evoluindo softwares robustos de Eric
Schulte**

Fábio Moreira Duarte

Sumário

1	Introdução	3
2	Trabalhos relacionados	3
2.1	Robustez e evolução em biologia	3
2.2	Computação evolutiva	4
2.3	Engenharia de software	4
3	Robustez mutacional	5
3.1	Robustez de software mutacional	5
3.1.1	Resultados experimentais	6
3.2	Causas da robustez da mutação	6
3.2.1	Níveis de representação: fonte vs compilado	6
3.2.2	Proveniência: evolução e engenharia	7
3.3	Correlatos de robustez mutacional	7
3.4	Robustez e evolução	7
3.5	Mutação e robustez de ambiente	7
4	Aplicações da robustez	8
4.1	Diverificação software	8
4.1.1	Compiladores e ligantes	8
4.1.2	Programa Atavismo e controle de veersão	8
4.2	Otimização de software	8
5	Criação de software	9
6	Plano de trabalho e linha do tempo	9
7	Conclusão	9

1 Introdução

Nos ultimos cinquenta anos, desenvolvedores de software selecionam, reutilizam e modificam ferramentas de desenvolvimento softwares eficientes e robusto, codigo e padrões de design. Atraves de um processo similar a seleção natural.

O objetivo da investigação é aumentar o entendimento de softwares existentes e ferramenta de desenvolvimento e manutenção. Focará-se em propriedade exibidas por sistemas complexos evolucionados e surgidas através de seleção natural e adaptação para sobrevivencia no mundo.

Hipotese propoe que propriedades similares levam para persistencia e replicação em ambos sistemas vivos e de engenharia. A proposta incluem funcionalidades robustas em um ambiente e habilidade de adaptar a novos ambientes.

Motivação: Nos ultimos cinquenta anos a produção e manutenção de software emergiu como uma industria importante. Envestigando as propriedades evolutivas dos sistemas de software, desafiara-se a sabedoria popular sobre a fragilidade de softwares, levando a novas ferramentas e tecnicas. Objetivo é aumentar a produção software robusto e reduzir o custo de manutenção.

Investigando robustez de software clareou-se propriedades fundamentais da robustez de sistemas evolucionados. Evolução digital engloba o design e analise de modelos computacionais e evolução biologica, permitindo evolução de quadros de tempo, controle e metricas não factiveis no local.

Oportunidade: Desenvolvimento em engenharia de software e biologia evolucionaria. Os dois campos se sobrepõem, com tecnicas de computação inspiradas em biologia sendo aplicadas em sistemas de manutenção de software e desempenho de software sendo discutidos em termo biologicos.

Visão geral: O plano de pesquisa divide-se em duas partes: focando na investigação de robustez mutacional de software e desenvolvimento de aplicações que alavanquem robustez de software.

2 Trabalhos relacionados

A pesquisa se da através estudos anteriores de sistemas biologicos e computacionais. Evolutividade e rebustez em biologia prove de conceitos e terminologias a serem investigadas em sistemas computacionais.

Utilizara-se as aplicações anteriores de computação para sistemas biologicos e processos biologicos para sistemas computacionais. Evolução digital demonstrou que o uso de sistemas de software clareou na natureza da robustez e biologia evolucionaria. Computação evolucionaria aplicou com sucesso computação analogica de seleção natural para o projeto, sistema de optimização e reparos.

2.1 Robustez e evolução em biologia

A habilidade de sistemas vivos de manter a funcionalidade através de varios ambientes e adaptar-se a um novo ambiente é incomparavel. Um problema é conhecer sobre a relação entre robustez e evolução em sistemas vivos.

Sistemas vivos incluem um genótipo e fenótipo. Genótipo é a informação genética que especifica um organismo vivo. O organismo resultante e sua interação com o mundo é o fenótipo.

Ambos genética e fenômeno tem tipos de robustez associadas. Genética é a habilidade do genótipo para produzir o mesmo fenótipo diante de perturbações no material genético. A robustez pode ser alcançada por diferentes caminhos e níveis. Nível mais baixo, aminoácidos importantes são representados como possível encodificação e codificação. Nível mais alto, funções vitais são degeneradas no sistema nervoso.

Mecanismos biológicos responsáveis por robustez mutacional, contribuem para robustez ambiental, ou a habilidade do fenótipo de manter-se diante de perturbações ambientais. Há correlação entre robustez genética e ambiental.

Robustez mutacional é um fator envolvido, evolução tende a aumentar a robustez da mutação de componentes biológicos importantes. Robustez mutacional é selecionada explicitamente para proteção contra mutação.

Robustez e evolução mutacional tem muitas faces. Robustez mutacional inibe evolução por qualquer mudança genética terá efeitos fenotípicos. Níveis merados de robustez mutacional aumentam diversidade genética da população habilitando evolução.

Um organismo robusto mutacional possui muito genótipos que mapeiam o mesmo fenótipo. Considera-se o espaço de genótipos ser espaço discreto de alta dimensão em que cada ponto é um genótipo, e o vizinhos de cada ponto são os genótipos que são alcançáveis através de uma única mutação. Regiões do espaço constituindo genótipos com mesmo fitness, são denominados espaço neutro. Populações tendem a espalhar em grande espaços neutros, acumulando variedade e material genético novo.

2.2 Computação evolutiva

Algumas propriedades da evolução seguintes, não são baseadas em observações diretas de sistemas biológicos, mas em modelos computacionais de sistema biológico. Em modelos computacionais de evolução digital a população evoluida representa o genótipo utilizando linguagem assembly em ambientes onde sua execução (fenótipo) determina sua reprodução ou extinção.

Evolução iluminada em sistemas biológicos gerou discernimento quanto as propriedades da linguagem de programação.

A evolução de programas de computador no processo de imitação da seleção natural é conhecida como programação genética.

2.3 Engenharia de software

A pesquisa pode usada em engenharia de software para enfatizar o desempenho aceitável para correção formal.

Beal e Sussman propoem um sistema para aumentar a robustez do software através de pré-processamento das entradas do programa. Sobre suposição que

operadores de software operam apenas em um subconjunto de entradas possíveis. Este sistemas de entrada “hallucination” aprimora a robustez de um unico sistema de reconhecimento de personagem.

Como as invariantes de aprendizagem e reforço do sistemas anteriores, na entrada do programa. Sistemas de visão clara aprendem invariantes no vestígio dos dados extraídos de execuções binárias utilizando Daikon. Quando ocorre violação por ataque, erro ou exploração, o sistema aplica um caminho de preservação invariante no binario que está executando, garantindo execução contínua.

As abordagens mencionadas são aplicadas em execução de sistemas de software ou fenítipo de software. Há técnicas aplicadas no software no código fonte pre-compilado, ou genótipo. São técnicas : perfuração de laço e botões dinâmicos. Software de perfuração de laço é compilado para uma simples representação intermediária, construtores de laço são encontrados e modificado, executando o laço algumas vezes, saltando todo n-esima execução do laço. Utilizada pra redução de energia e custo de execução, mantendo limites probabilísticos de correteude esperada.

Metodos de programas de evolução, reparo e otimização, aplicam técnicas de computação evolutiva no código fonte de projetos de software existentes. Testa as funções de fitness usadas para avaliar os efeitos da transformação do programa.

3 Robustez mutacional

Definiu-se a robustez da mutação de software como porcentagem de softwares mutantes equivalentes ao programa original. Software é a robustez da mutação de seu código fonte e assembly. Possuindo multiplas implicações para natureza do software e potenciais novas ferramentas de software.

3.1 Robustez de software mutacional

Espaço sintático será espaço do texto do código fonte de um programa, espaço semântico é o espaço da funcionalidade do comportamento do programa. Compilação será a função do espaço da sintaxe para o espaço da semântica. Dado um programa aplica-se mutações semanticas e observa-se a semantica dos mutações resultantes.

A comunidade de teste de mutação usa operadores de mutação projetados para produzir mutantes de programas de linguagem. Assume que todos mutantes são defeitosos ou equivalentes. Um indice adequado de mutação acessa a qualidade de conjuntos com porcentagem de mutantes equivalentes que falham em teste.

Não permitindo mutantes não defeitosos ou não equivalente, os testes de mutação ignoram variantes funcionais

Robustez mutacional de software é a porcentagem de mutações aleatorias dado uma instância de software que deixa as funcionalidades observadas inal-

teradas. Assume-se que mutantes que satisfazem o teste de saída também satisfazem a especificações, determina-se fitness da mutação com softwares com conjuntos de teste existente. Apenas variantes compiladas com sucesso e que passem por todos os teste são consideradas neutras. Operadores de mutação restringem-se a partes do programa com os teste de saída, para garantir que mutações tenham efeito na execução do software.

3.1.1 Resultados experimentais

Testou-se a robustez de 22 programas. O algoritmo de ordenação completou os teste de saída garantindo que cada estado, ramo, e intrusão em assembly seja avaliada pelo teste de saída.

Para cada programa, gerou-se pelo menos 200 variantes unicas do programa, utilizando operadores de mutação (inserção, remoção e troca).

3.2 Causas da robustez da mutação

Há numerosas propriedades de sistemas biologicos para causar altos niveis de robustez mutacional.

O mapeamento do DNA para aminoácidos evoluiu de tal forma que sequencias similares de DNA criam proteínas similares. O mapeamento de sequencias de RNA para funções, produziu propriedades como continuidade local e alcance global.

3.2.1 Niveis de representação: fonte vs compilado

Compilação e conexão de codigos do programa é analogo a processos biologicos de transições e translações de DNA através de genótipos biologicos codificados no DNA translado em estruturas de proteínas, onde iterações dão origem a fenótipos.

O processo biologico de transcrição e translação contribuem para robustez mutacional de organismos. Por hipotese o mesmo é verdade em software. Sendo esse o caso, compiladores e conectores contribuem para robustez mutacional do software.

Testou-se a proposição por comparações de robustez da mutação em codigos fontes, codigo assembly e arquivos ELF.

Ameaças a validação: a diferença em representações entre niveis de representação, requer diferentes operadores de mutação serem utilizados em cada nivel. Medições de robustez mutacional é tao quanto operadores mutacionais utilizados como representações operadas. Cuidados deve ser tomados para garantir que os operadores são similares e os efeitos sejam entendidos, garantindo que diferentes observações são diferentes na robustez inerente.

Tendo diferenças significantes na robustez do programa entre niveis, o proximo passo é investigar os mecanismo especificos do compilador e conexões de robustez.

3.2.2 Proveniência: evolução e engenharia

Produtos de processo evolucionário são mais robustos a mudanças através de processos evolucionários, e acessível a aprimoramentos através dos processos. Se os resultados generalizam o software, os artefatos programados usando técnicas evolutivas serão mais robustos. Propõe-se teste da hipótese por comparações de três tipos de artefatos.

1. Programados totalmente por engenheiros humanos.
2. Programados inicialmente por engenheiros humanos e evoluídos incrementalmente.
3. Programados inteiramente por processos evolutivos.

O terceiro tipo do software requer a nova evolução para ser examinado, para restringir a investigação à linguagem simples e algoritmos acessíveis ao desenvolvimento através de processos evolutivos.

3.3 Correlatos de robustez mutacional

Investigando os casos possíveis de robustez mutacional em software, procura-se por possíveis efeitos ou correlatos de robustez mutacional. Propõe-se o estudo da relação entre robustez mutacional e evolutividade de software e a relação entre mutação e robustez do ambiente.

3.4 Robustez e evolução

Em sistemas biológicos robustez mutacional e evolutividade estão ligadas. Robustez mutacional permite mutações neutras acumularem na população, aumentando a diversidade. Excessiva robustez mutacional reduz muitas mutações neutras, inibindo seleção evolutiva de mutantes benéficos.

Propõe-se o estudo dos efeitos de vários níveis de robustez mutacional. Podendo indicar qual nível de robustez mutacional é desejável no software e quais casos aumentar a robustez mutacional seria benéfico ou prejudicial.

Tais experimentos requerem métricas de evolutividade e robustez mutacional.

3.5 Mutação e robustez de ambiente

Mecanismos biológicos são considerados comuns na genética e robustez mutacional. Proteja vários níveis de entrada independente do caso da variante.

Propõe-se investigar qual robustez mutacional é possivelmente correlatada com robustez do ambiente. Robustez ambiental de software é habilidade de executar com sucesso em um ambiente computacionais.

4 Aplicações da robustez

A maleabilidade de software torna plausível numerosas técnicas para diversificação de software.

4.1 Diverficação software

A hipótese é que espaços neutros de sistemas de software pode ser explorados para gerar variantes de software.

4.1.1 Compiladores e ligantes

Pedaços de código fonte pode ser usados para gerar numerosos programas distintos executáveis. O processo de compilação determina características do executável final, não especificados pelo código fonte.

Expressões diversas de partes do código fonte podem ser usadas para buscar uma população evolutiva de diversas variantes do programa. Reprodução neutra da população pode misturar e combinar características entre executáveis.

Propoe o uso da variedade das distintas compilações, buscando processos evolutivos com variantes de entrada do programa.

4.1.2 Programa Atavismo e controle de veersão

Sistemas biológicos retem informação codificada de fenótipos, de tal forma que podem ser acessados, através de mutação genética espelhada no processo atavismo. Desenvolveu-se um método automatizado de programa atavismo utilizando armazenamentos de informações em repositórios de controle de versão. Controle de versão de informação será codificada em representação evolutiva.

A informação armazenada no controle de versão do repositório tem translações implícitas e explícitas de sistemas de memória. Pacotes alternativos definem implementações alternativas para porções do programa. O histórico de versão pode ser visto como um conjunto de pacotes, podendo ser armazenada em programas representativos de memória. Podem ser armazenada e utilizada com população futuramente.

O trabalho terá foco inicial em representação de memória, provendo suporte para material genético não codificado.

Os experimentos serão limitados a programas como históricos com edição considerável e em caos de reparação de programas.

4.2 Otimização de software

Durante compilação e ligação, propriedades não funcionais de software como tempo de execução e tamanho de execução podem ser otimizadas.

Técnicas atuais exclusivamente em operadores pode ser provadas formalmente por programas de semânticas. Utilizando as suites de teste devem perder baseados na definição do programa, espera-se evoluir variantes de programa neutras não alcançáveis de operações de através de preservação semântica. Multi

objetivos podem ser usados para evoluir variantes de software semanticamente neutros.

5 Criação de software

A evolução de diversas populações de programas únicos levanta questões de identidade de software.

É possível que indivíduos de regiões separadas possam ser combinados para formas híbridas de implementação contendo material genético de ambos os programas.

Recombinações de programa sem ancestrais em comum foi observada, realizada em nível de objeto.

Propõe o uso de testes únicos para desenvolver populações neutras. Combinar indivíduos utilizando operações de cruzamento.

6 Plano de trabalho e linha do tempo

1. Reparo de assembly
2. Artigos pequenos ASE
3. Representação ELF
4. Robustez mutacional
5. Reparos embutidos
6. Comparações de representação
7. Engenharia/Evolução Rb.
8. Fontes de robustez
9. Ambiente/Mutação Rb.
10. Otimização
11. Diversidade
12. Robustez mutacional

7 Conclusão

Sistemas de engenharia de software são robustos, limitados a sistemas biológicos. Propõe uma investigação da robustez de sistemas de software e numerosas ferramentas aproveitando a robustez.

A investigação isola as fontes de robustez no software, determinando os efeitos da robustez na evolutividade do software, buscando correlações entre robustez de software.