



Universidade Federal de Uberlândia
Faculdade de Computação



Fábio Moreira Duarte

Aplicação de algoritmos meta heurísticos na escolha de
instruções no back-end do compilador, utilizando abordagem
distribuída

Uberlândia
2017

Fábio Moreira Duarte

Aplicação de algoritmos meta heurísticos na escolha de instruções no
back-end do compilador, utilizando abordagem distribuída

Projeto de pesquisa apresentado como exigência
para elaboração do Trabalho de Conclusão de
Curso do Bacharelado em Ciência da Computa-
ção da Universidade Federal de Uberlândia.

Orientador: Alexsandro Santos Soares

Uberlândia
2017

Abreviações

BCC Bacharelado em Ciência da Computação.

FACOM Faculdade de Computação.

M Material.

N NÃO.

NP Não Polinomial.

P Pergunta.

R Resposta.

S SIM.

UFU Universidade Federal de Uberlândia.

Sumário

Abreviações	2
1 Introdução	4
1.1 Questão de pesquisa	4
1.2 Hipótese	4
1.3 Objetivo	4
1.4 Justificativa	5
2 Guias	5
3 Metodologia	5
3.1 Tipo de pesquisa	5
3.2 Tipos de dados	6
3.3 Aquisição de dados	6
3.4 Análise de dados	6
3.5 Limitações	6
4 Protocolo	6
4.1 Revisão do cronograma	6
5 Revisão bibliográfica	7
5.1 Busca por fontes de pesquisa	7
5.2 Critérios de Inclusão e Exclusão de artigos	8
5.3 Levantamento sistemático	9
5.4 Busca por trabalhos semelhantes	9
6 Restrições	11

1 Introdução

Este material tem como objetivo documentar os métodos utilizados para a condução da monografia do estudante de bacharelado em Ciência da Computação, Fábio Moreira Duarte, no campos da UFU, Brasil, no ano de 2017, sobe orientação do professor Alexsandro Santos Soares.

Para a condução do projeto fez-se necessário, compreender os fundamentos e técnicas para formulação de um artigo ou estudo de maneira científica, foram utilizados três livros como guias que serão apresentados no capítulo 2. Após leitura e estudo, aplicou-se os ensinamentos obtidos.

Escolhido o tema, iniciou-se no capítulo 5 a revisão bibliográfica, definindo as fontes de informação, através do levantamento sistemático construiu-se as questões de pesquisa, trabalhos relacionados.

1.1 Questão de pesquisa

Aplicar algoritmos meta heurísticos para otimização na escolha de instruções em um ambiente distribuído, pode retornar um resultado ótimo, em tempo suportável?

1.2 Hipótese

Aplicar algoritmos meta heurísticos de maneira distribuída para otimização na escolha de instruções em um compilador, gera um resultado semelhante ou melhor em um tempo suportável.

1.3 Objetivo

O objetivo geral desse trabalho é comparar algoritmos meta heurístico que possam ser implementados de forma distribuída para otimização na escolha de instruções em um compilador:

Etapas do objetivo:

1. Implementar diversos algoritmos de meta heurísticos de tal forma que possam ser executados de maneira distribuída.
2. Substituir o algoritmo de otimização do compilador pelos algoritmos meta heurísticos.
3. Realizar experimentos utilizando os algoritmos meta heurístico e o algoritmo original do compilador.
4. Realizar os experimentos aumentando a quantidade de máquinas utilizadas, anotando padrões
5. Confrontar os dados dos experimentos.
6. Verificar se a hipótese é verdadeira.

1.4 Justificativa

Seleção de instruções, alocação de registradores e escalonamento de instruções são problemas de compilação críticos para qualidade [3, 10]. Com intuito de solucionar tais problemas, técnicas de otimização são aplicadas em numerosas atividades em engenharia de software [2].

Varias otimizações e técnicas de busca podem ser usadas, busca local, recozimento simulado, algoritmos genéticos e programação genética. [2]. “Contudo, não importa qual técnica de busca é aplicada, é a função fitness que capta a informação crucial” Harman 2007, page 01, texto traduzido do inglês. Fitness decide quão próximo do resultado valor esta.

Devido a complexidade de grande parte das otimizações serem NP-completo e o grande numero de combinações possíveis [3].

2 Guias

Devido a inexperiências com trabalhos anteriores, guias de condução de trabalhos de pesquisa foram sugeridos pelo orientador.

O primeiro o livro Metodologia de pesquisa para ciência da computação de Wazlawick 2015 esclareceu as diversas formas de ciências existentes dentre formais e empíricas, exatas e inexatas e outras, listando as etapas para criação de trabalho de pesquisa, principalmente no capítulo 6 onde apresentou o que é o objetivo de pesquisa, revisão bibliográfica, objetivo, método de pesquisa, a hipótese de pesquisa, justificativa de hipótese, os resultados esperados, a limitação do trabalho e discussão. Dentre outros tópicos. O livro relata de maneira superficial seus assuntos.

O segundo livro Redação científica: a prática de fichamentos, resumos, resenhas de Medeiros, 2000. Sugeriu dicas para aumentar a eficiência de leitura, absorção de conhecimento e uso do tempo, apresentando técnicas para leitura e resumo.

O terceiro livro Evidence-based software engineering and systematic reviews de Kitchenham et al., 2015, aprofundou na construção de trabalhos de pesquisa especialmente em engenharia de software, definiu diversos assuntos apresentados no livro de Wazlawick.

Evidence-based software engineering and systematic reviews explicou de maneira clara assuntos que terão grande impacto no artigo, como: revisão bibliográfica, revisão sistemática e mapeamento de estudos, apresentando exemplos para melhor entendimento.

3 Metodologia

Para realização da monografia, definiu-se a metodologia empregada.

3.1 Tipo de pesquisa

Analisando os metodos de pesquisa existentes [5], optou-se por uma pesquisa bibliográfica, fazendo uso de artigos e apresentações para conhecer a situação atual na área de compiladores especialmente na parte de otimização, fez se o levantamento sistemático 5.3. Foram encontrados utilizando os metodos de busca 5.1, artigos 5.2 que descreviam a área otimização em compiladores.

3.2 Tipos de dados

Como o objetivo é otimização, os dados serão valores de registro do consumo de tempo, memória e energia. O tempo de início e fim da execução, a quantidade de memória utilizada durante a execução e componentes utilizados, calculando com dados dos fabricantes o custo de energia estimado.

3.3 Aquisição de dados

Os dados serão coletados através de experimentos realizados em diversas máquinas, utilizando algoritmos simples e diferentes. Os algoritmos escolhidos serão compilados utilizando os métodos meta-heurísticos escolhidos e o método original utilizado pelo compilador. Os dados serão coletados após execução das versões compiladas, em máquinas com configurações iguais ou semelhantes, desabilitando o máximo de funcionalidades para redução de erro.

3.4 Análise de dados

Após a coleta, os dados serão comparados, os resultados obtidos utilizando os algoritmos meta-heurísticos escolhidos contra os resultados obtidos na função original do compilador. Após análise será decidido se a hipótese é verdadeira ou não.

3.5 Limitações

Limitações que podem interferir nos resultados.

- Operações do sistema operacional que não podem ser desabilitadas, podem produzir ruído no tempo de execução.
- Máquinas com dispositivos embarcados mais eficientes ou métodos diferentes para execução da intrusão, podem alterar os dados de energia e memória e tempo.

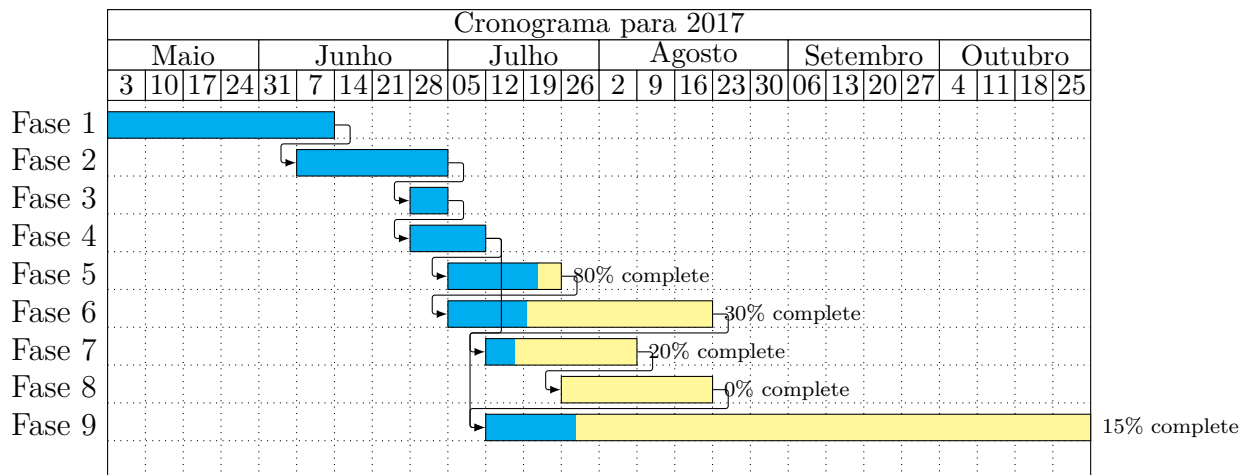
4 Protocolo

A revisão bibliográfica é uma parte importante para o processo de pesquisa, todo processo realizado na revisão bibliográfica está especificado no capítulo [5](#).

4.1 Revisão do cronograma

Fase: 1 Leitura de livros guias de desenvolvimento de artigo científico em engenharia de software

- Fase: 2 Desenvolvimento do protocolo
- Fase: 3 Identificação de ferramentas de busca
- Fase: 4 Busca por literatura semelhante e relevante
- Fase: 5 Avaliação qualitativa do material encontrado
- Fase: 6 Leitura do material encontrado
- Fase: 7 Busca por literatura suplementar
- Fase: 8 Avaliação qualitativa do material encontrado
- Fase: 9 Escrita da monografia



As datas escolhidas representam as reuniões com orientador que ocorriam toda quarta-feira

5 Revisão bibliográfica

Seguindo os conhecimentos obtidos com os guias na seção guias, começou-se o desenvolvimento da revisão bibliográfica. A revisão tem como objetivo obter uma pesquisa ampla sobre o tema de tópico, no caso “Aplicação de algoritmos meta heurísticos na escolha de instruções no back-end do compilador, utilizando abordagem distribuída” .

5.1 Busca por fontes de pesquisa

Para obtenção do material necessário para o trabalho de pesquisa, decidiu-se utilizar buscadores especializados, com intuito de maximizar a obtenção de materiais relevante ao tema.

Fez-se varias pesquisas utilizando ferramentas de busca automatizada: Google, Yahoo e Bing.

Palavras chaves: As seguintes palavras-chaves foram usadas, aplicando combinações de AND e OR:

- Monograph
- Source
- Tool
- Online
- Academy
- Popular
- Writing
- Free
- Research
- Development

Foram encontradas diversas ferramentas, listadas na tabela 5.1, aplicou-se os critérios de inclusão e exclusão listados na legenda da tabela. A exclusão ocorreu para todos aqueles que obtiver uma nota inferior a 4 de uma escala de 1 a 5.

Tabela 5.1 – Fontes de pesquisa

Critérios de exclusão/inclusão	<i>R1</i>	<i>R2</i>	<i>R3</i>	<i>R4</i>	<i>R5</i>	<i>Nota</i>
Academic Index	S	N	N	N	N	1
Google Scholar	S	S	N	S	S	4
Microsoft Academic	S	S	S	S	N	4
RefSeek	S	N	N	N	N	1
Virtual LRC	S	N	N	N	S	2
Digital Library of the Commons Repository	S	N	S	N	S	3
iSEEK Education	S	N	N	N	S	2
Internet Public Library	S	N	N	N	N	1

Respostas possíveis:

N: não atende o requisito. Peso 0 ponto.

S: atende o requisito. Peso 1 ponto.

R1: Disponibiliza o material de pesquisa de maneira gratuita para estudante?

R2: Permite filtragem por período?

R3: Permite filtragem por autor?

R4: Permite ordenação por citação?

R5: Possui sistema de classificação?

A nota é dada pela soma dos pesos das perguntas respondidas.

Excluindo as ferramentas com nota inferior a 4, ficamos apenas com Google Scholar e Microsoft Academic. Serão as fontes de material para a pesquisa.

5.2 Critérios de Inclusão e Exclusão de artigos

Serão considerados apenas os estudos que atenderem os seguintes critérios:

- Disponíveis em inglês ou português
- Estudos completos

- Realizados entre o período de 2000 a 2016
- Caso implemente algum algoritmo, o código fonte deve estar disponível

5.3 Levantamento sistemático

Com intuito de desenvolver as questões para seleção buscou-se as perguntas que outros pesquisadores determinaram importantes para o tema.

Fez-se uma busca para encontrar levantamentos na área de compiladores, otimização.

Palavras-chaves: As seguintes palavras-chaves foram usadas, aplicando combinações de AND e OR:

- Fitness
- Survey
- Software
- Enginnering
- Optimization
- Compiler
- Choose
- Metaheuristic

Tabela 5.2 – Levantamento sistemático

Levantamentos encontrados	
The current state and future of search based software engineering	[2]
Survey on Instruction Selection An Extensive and Modern Literature Review	[3]
Survey of HDL Compiler Optimization Techniques	[4]
Survey on Combinatorial Register Allocation and Instruction Scheduling	[10]

Após análise dos artigos encontrados, descobriu-se os principais problemas da área de otimização em compiladores, sendo eles: Seleção de instruções, alocação de registradores e escalonamento de instruções [3, 10]. Com a grande parte dos metodos de otimização disponiveis são NP-completo [3]

5.4 Busca por trabalhos semelhantes

Utilizando as ferramentas de busca selecionadas. Iniciou-se a busca por publicações na área de interesse, procurando por materiais correlacionados, com intuito de aprofundar no assunto.

Os artigos serão lidos conforme sua nota de maneira decrescente.

Seguindo as orientações dos guias, é necessário conhecer o que já foi realizado na área de estudo. Analisando o tema escolhido “Aplicação de algoritmos meta heurísticos na escolha de instruções no back-end do compilador, utilizando abordagem distribuída”, as seguintes questões foram construídas.

Perguntas:

P1 Quais são as técnicas de otimização disponíveis pelo compilador?

P2 Quais os algoritmos mais utilizados para otimização?

P3 É possível aplicar algoritmos meta heurísticos em ambientes distribuídos?

Sendo o tema “Aplicação de algoritmos meta heurísticos na escolha de instruções no back-end do compilador, utilizando abordagem distribuída”, foram utilizadas as seguintes palavras-chaves para obter os estudos já realizados na área, assuntos semelhantes, outros métodos de otimização e o uso de meta heurística em outras áreas.

Palavras-chaves: As seguintes palavras-chaves foram usadas, aplicando combinações de AND e OR:

- | | |
|-----------------|---------------|
| • Compiler | • Application |
| • Optimizations | • Distributed |
| • Metaheuristic | • System |

Tabela 5.3 – Trabalhos semelhantes

Trabalhos encontrados		<i>P1</i>	<i>P1.1</i>	<i>P2</i>	<i>P2.1</i>	<i>P3</i>	<i>P3.1</i>	<i>Nota</i>
Central force optimization on a...	[1]	S	-	S	-	N	S	2.5
Genetic Instruction Scheduling and...	[6]	S	-	S	-	N	-	2
Evaluation of algorithms for local re...	[7]	?	?	?	?	?	?	?
Uma solução para o problema de...	[8]	S	-	N	S	N	N	1.5
Register Allocation with Graph Co...	[9]	S	-	N	S	N	N	1.5
Exploração de sequências de otimi...	[11]	?	?	?	?	?	?	?
Post-compiler Software Optimiza...	[13]	N	N	N	S	N	N	0.5
Ant colony optimization for continu...	[14]	S	-	N	S	N	N	1.5
Extending Traditional Graph-...	[16]	S	-	N	S	N	N	1.5

Respostas possíveis:

N: O artigo não responde a pergunta. Peso 0 ponto.

S: O artigo responde a pergunta. Peso 1 ponto para perguntas primárias e 0.5 ponto para perguntas secundárias.

-.: A pergunta primária já foi respondida. Peso 0 ponto.

Perguntas:

P1, P2 e P3 são perguntas específicas sobre o tema.

P1.1, P2.1 e P3.1 são perguntas secundárias podendo ser respondidas, caso sua pergunta primária correspondente tenha sido respondida de maneira negativa.

A nota é dada pela soma dos pesos das perguntas respondidas.

Perguntas:

P1 Faz uso de algoritmos meta heurísticos?

P11. Faz uso de meta heurística?

P2 Busca otimização na escolha de instruções?

P21. Busca otimização?

P3 Aplica abordagem distribuída?

P31. Possui abordagem paralelizada?

6 Restrições

Não foram encontrados estudos que respondessem todas as questões primárias especificadas na tabela 5.3. Sendo assim não foi encontrado nenhum estudo com tema equivalente ao do proposto pelo projeto para material de apoio.

O conhecimento de apenas dois idiomas, limitou o pesquisador quanto a obtenção de trabalhos.

Foram encontrados bons levantamentos sobre a situação de otimização em compiladores, porem todas as versões encontradas não pertenciam ao período de tempo de inclusão e exclusão, sendo assim foram descartadas.

Referências Bibliográficas

- [1] Robert C Green, Lingfeng Wang, Mansoor Alam, and Richard A Formato. Central force optimization on a gpu: a case study in high performance metaheuristics. *The Journal of Supercomputing*, 62(1):378–398, 2012.
- [2] Mark Harman. The current state and future of search based software engineering. In *2007 Future of Software Engineering*, pages 342–357. IEEE Computer Society, 2007.
- [3] Gabriel Hjort Blindell. Survey on instruction selection: An extensive and modern literature review, 2013.
- [4] M Joseph, Narasimha B Bhat, and K Chandra Sekaran. Survey of HDL compiler optimization techniques. *International Journal on Information Processing*, 1(1):51–62, 2007.
- [5] Barbara Ann Kitchenham, David Budgen, and Pearl Brereton. *Evidence-based software engineering and systematic reviews*, volume 4. CRC Press, 2015.
- [6] Fernanda Kri and Marc Feeley. Genetic instruction scheduling and register allocation. In *Computer Science Society, 2004. SCCC 2004. 24th International Conference of the Chilean*, pages 76–83. IEEE, 2004.
- [7] Vincenzo Liberatore, Martin Farach-Colton, and Ulrich Kremer. Evaluation of algorithms for local register allocation. *CC*, 99:137–152, 1999.
- [8] Carla Négre Lintzmayer, Mauro Henrique Mulati, and Anderson Faustino da Silva. Uma solução para o problema de alocação de registradores baseada em meta-heurísticas.
- [9] Carla Negre Lintzmayer, Mauro Henrique Mulati, and Anderson Faustino da Silva. Register allocation with graph coloring by ant colony optimization. In *Computer Science Society (SCCC), 2011 30th International Conference of the Chilean*, pages 247–255. IEEE, 2011.
- [10] Roberto Castaneda Lozano and Christian Schulte. Survey on combinatorial register allocation and instruction scheduling. *arXiv preprint arXiv:1409.7628*, 2014.
- [11] Luiz Gustavo Almeida Martins. *Exploração de sequências de otimização do compilador baseada em técnicas híbridas de mineração de dados complexos*. PhD thesis, Universidade de São Paulo, 2016.
- [12] João Bosco Medeiros. *Redação científica: a prática de fichamentos, resumos, resenhas*. Editora Atlas SA, 2000.
- [13] Eric Schulte, Jonathan Dorn, Stephen Harding, Stephanie Forrest, and Westley Weimer. Post-compiler software optimization for reducing energy. In *ACM SIGARCH Computer Architecture News*, volume 42, pages 639–652. ACM, 2014.
- [14] Krzysztof Socha and Marco Dorigo. Ant colony optimization for continuous domains. *European journal of operational research*, 185(3):1155–1173, 2008.

- [15] Raul Wazlawick. *Metodologia de pesquisa para ciência da computação*, volume 2. Elsevier Brasil, 2015.
- [16] Shengning Wu and Sikun Li. Extending traditional graph-coloring register allocation exploiting meta-heuristics for embedded systems. In *Natural Computation, 2007. ICNC 2007. Third International Conference on*, volume 4, pages 324–329. IEEE, 2007.