

Uma hiper-heurística de seleção baseada em decomposição para estabelecer sequências de módulos para o teste de software

Vinicius Renan de Carvalho
Orientadora: Profa. Dra. Silvia Regina Vergilio

DInf - Universidade Federal do Paraná (UFPR)

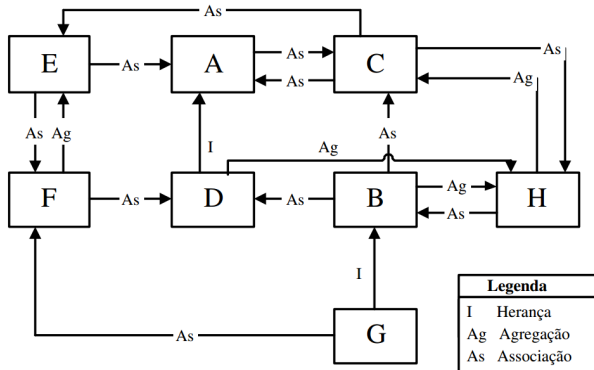
2015

Conteúdo Programático

- 1 Contexto
- 2 Motivação
- 3 Justificativa
- 4 Objetivos do trabalho
- 5 Algoritmos Baseados em Decomposição
- 6 Hiper-heurística
- 7 Hiper-heurísticas aplicadas à SBSE
- 8 HITO
- 9 Contribuições
- 10 Estudo Experimental
- 11 Conclusões

- O problema de estabelecimento de sequências de módulos para o teste de integração (*Integration Test Order problem (ITO)*) visa a determinar uma ordem em que os módulos de um sistema devem ser integrados e testados, com o intuito de reduzir o esforço destas tarefas;
- Na maioria das vezes os módulos não são integrados de uma vez;
- Algumas vezes *stubs* deve ser implementados;
- O problema é representado como um grafo direcionado.

Representação do Problema



Abordagens para resolver o problema

- Algoritmos baseados em técnicas de grafos;
- Abordagens Mono-objetivos
 - Algoritmo Genético;
- Abordagens Multi-objetivo
 - PACO
 - NSGA-II
 - MTabu
 - SPEA2
 - PAES
 - MOCAITO
- Hiper-heurísticas
 - HITO
 - MOCAITO-HH

Motivação

- O uso de hiper-heurísticas desperta interesse da comunidade de SBSE, mas ainda são poucos os trabalhos que tratam deste tópico;
- O uso de hiper-heurísticas aplicadas ao problema ITO tem obtido bons resultados, especialmente as que trabalham com o algoritmo de seleção *Choice Function* (CF);
- Li et al. obtiveram bons resultados ao aplicar em seus trabalhos FRRMAB em conjunto com o MOEA/D-DRA;
- Algoritmos baseados em decomposição podem auxiliar na melhora de resultados.

Justificativa

- O uso de algoritmos baseados em decomposição pode contribuir para a melhora dos resultados existentes, pois estes algoritmos têm obtido bons resultados na literatura;
- A HITO não foi projetada tendo em mente algoritmos baseados em decomposição;
- O FRRMAB tem apresentado melhores resultados do que outros algoritmos derivados do MAB. Principalmente por usar os valores de FRR (Fitness Rate Ranking);
- O algoritmo de seleção CF pode-se beneficiar do uso dos valores FRR.

Objetivos do trabalho

- Propor a hiper-heurística HITO-DA (*Hyper-heuristic for the Integration and Test Order Problem using Decomposition Approach*);
- Propor o algoritmo de seleção FRRCF (*Fitness Rate Rank Choice Function*) para trabalhar com a HITO-DA, pois a HITO obteve bons resultados com a CF.

Multi-Objective Evolutionary Algorithm based on Decomposition (MOEA/D)

- Decompõe um problema multiobjetivo em subproblemas para que sejam trabalhados individualmente;
- Cada subproblema possui uma solução associada;
- Mantém um conjunto de vetores de peso ($\Lambda = \{\lambda^1, \dots, \lambda^N\}$) e um conjunto de pontos ideais ($z^* = \{z_1^*, \dots, z_m^*\}$);
- Os subproblemas também usam informações da vizinhança na busca de soluções;
- É realizado um determinado número de substituições de soluções;
- Uso de uma função de agregação para o cálculo do valor de *fitness*, como soma ponderada ou Tchebycheff:

$$g^{te}(x|\lambda, z^*) = \max\{\lambda_i | f_i(x) - z_i^* |\}$$
$$1 \leq i \leq m$$

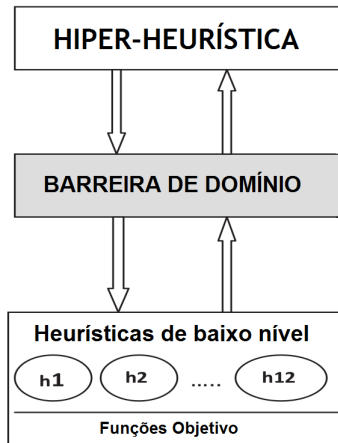
Multi-Objective Evolutionary Algorithm based on Decomposition with Dynamical Resource Allocation (MOEA/D-DRA)

- No MOEA/D, todos os subproblemas são tratados igualmente e recebem o mesmo esforço computacional;
- O MOEA/D-DRA aloca diferentes recursos computacionais para os diferentes subproblemas de acordo com a utilidade;
- A utilidade leva em consideração a melhora obtida por um subproblema ao longo do processo evolutivo.

Conceito de Hiper-heurística

- Segundo Cowling et al. (2001), hiper-heurísticas são heurísticas que escolhem heurísticas.
- Segundo Burke et al.(2013), hiper-heurísticas são:
 - (a) metodologias de seleção de heurísticas;
 - (b) metodologias de geração de heurísticas.

Barreira de domínio



Choice Function (CF)

- Adaptativamente classifica cada LLH com relação a uma pontuação;
- Considera:
 - Desempenho;
 - Tempo de espera desde sua última utilização.
- Possui dois parâmetros, α e β .

$$op = \operatorname{argmax}_{i=1\dots K}(\alpha f_1(h_i) + \beta f_2(h_i)) \quad (2)$$

Multi Armed Bandit (MAB)

- Proposto para solução do problema MAB da teoria de jogos;
- Atribui uma LLH para cada um dos braços;
- Cada LLH i é associada a:
 - Uma recompensa empírica q_i (como por exemplo, a média das melhoras obtidas);
 - Um intervalo de confiança Nt_i , representado pelo número de vezes que a i -ésima LLH foi acionada.
- Algumas versões do MAB usam o conceito de janela de tempo.

$$op = \operatorname{argmax}_{i=1\dots K} \left(q_i + C * \sqrt{\frac{2 * \ln \sum_{j=1}^K Nt_j}{Nt_i}} \right) \quad (3)$$

Fitness Rate Ranking Multi Armed Bandit (FRRMAB)

- Derivação do MAB;
- Usa valores de FRR (*Fitness Rate Ranking*) como recompensa q ;
- Os valores de FRR são calculados de forma que a influência da LLH que tenha os melhores resultados possa ser incrementada.

FRRMAB - Atribuição de Crédito (Credit Assignment)

Algoritmo 1: Pseudocódigo da atribuição de crédito do FRRMAB

```
1 Entrada: SlidingWindow, D, K
2 Saída: FRR, Nt
3 início
4     Atribuir 0 para cada elemento em Reward;
5     Atribuir 0 para cada elemento em Nt;
6     para Item in SlidingWindow faça
7         FIR = Item.getFIR();
8         op = Item.getIndexOp();
9         Rewardop = Rewardop + FIR;
10        Ntop ++;
11    fim
12    Classificar vetor Reward em ordem
13    decrescente;
14
15    para op ← 1 até K faça
16        Rankop = Reward.getOpPosition(op);
17    fim
18    para op ← 1 até K faça
19        Decayop =  $D^{Rank_{op}} * Reward_{op}$ ;
20    fim
21    DecaySum =  $\sum_{op=1}^K Decay_{op}$ ;
22    para op ← 1 até K faça
23        FRRop = Decayop / DecaySum;
24    fim
25    retorna FRR;
```


FRRMAB - Recompensa (Reward)

$$Decay_{op} = D^{Rank_{op}} * Reward_{op} \quad (4)$$

$$DecaySum = \sum_{op=1}^K Decay_{op} \quad (5)$$

$$FRR_{op} = \frac{Decay_{op}}{DecaySum} \quad (6)$$

Algoritmo 2: Pseudocódigo do método de seleção do FRRMAB

```
1 Entrada: FRR, C, K, Nt
2 Saída: op
3 início
4   se Existem LLHs ainda não selecionadas então
5     op=Selecionar aleatoriamente alguma LLH ainda não selecionada;
6   fim
7   senão
8      $op = \operatorname{argmax}_{i=1\dots K} (FRR_i + C * \sqrt{\frac{2 * \ln \sum_{j=1}^K Nt_j}{Nt_i}});$ 
9   fim
10  retorna op;
11 fim
```

FRRMAB - Fitness Improvement Rate (FIR)

O FIR é o somatório das melhoras proporcionadas pelos filhos em relação aos pais ao aplicar uma dada LLH op . Nesta Equação $F(p)$ é o valor de *fitness* do pai p e $F(c)$ é o valor de *fitness* do filho c .

$$n = \frac{(F(p) - F(c))}{F(p)} \quad (7)$$

$$FIR_{op} = FIR_{op} + n, n > 0$$

Hiper-heurísticas aplicadas à SBSE

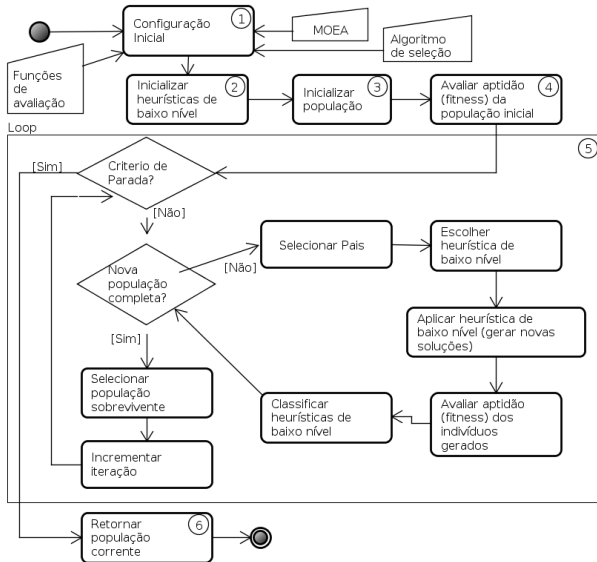
Segundo Harman et al. o uso de hiper-heurísticas desperta interesse da comunidade de SBSE, mas ainda são poucos os trabalhos que tratam deste tópico.

- Jia et al. introduziram uma HH para aprender e aplicar estratégias de teste combinatorial;
- Basgalupp et al. propuseram uma HH para a geração de algoritmos que criam árvores de decisão, a serem utilizadas na predição de esforço de software.
- Kumari e Srinivas propuseram uma HH para trabalhar com o problema de agrupamento de módulos;
- Carvalho et al. propuseram uma HH para buscar soluções para o problema ITO tratando meta-heurísticas como operadores.

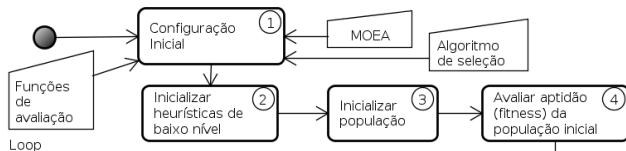
Hyper-heuristic for the Integration and Test Order Problem (HITO)

- Hiper-heurística de aprendizado online;
- Trabalha com a seleção de LLHs perturbativas (operadores de mutação e cruzamento);
- Busca selecionar a melhor LLH durante o processo de busca;
- Trabalha buscando soluções para o problema ITO.

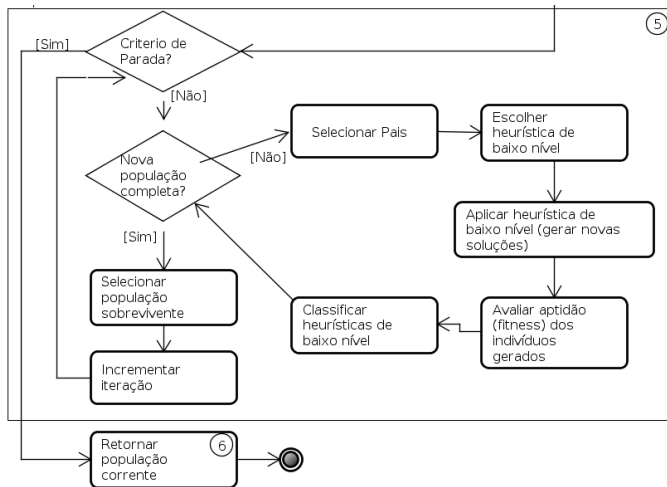
HITO - Fluxo



HITO - Fluxo



HITO - Fluxo

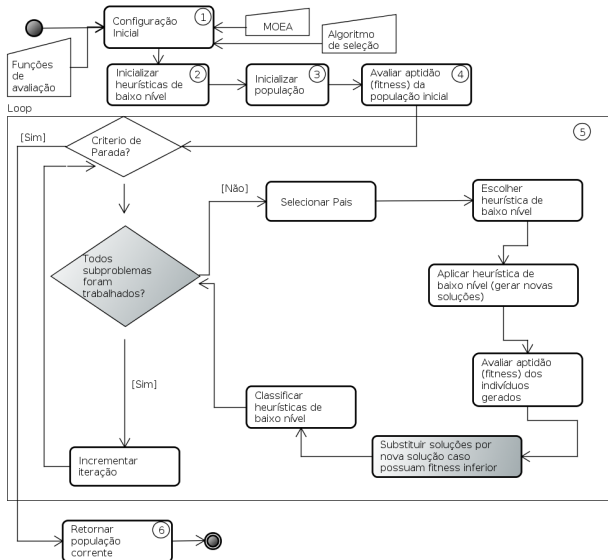


- Objetivos do problema
 - Número de operações referenciadas entre módulos.
 - Número de atributos referenciadas entre módulos.
 - Número de tipos de retorno referenciados entre módulos.
 - Número de tipos de parâmetros referenciados entre módulos.
- Meta-heurísticas
 - NSGA-II
 - SPEA2
- Algoritmos de seleção
 - MAB
 - CF
- Função de avaliação
 - Métrica r

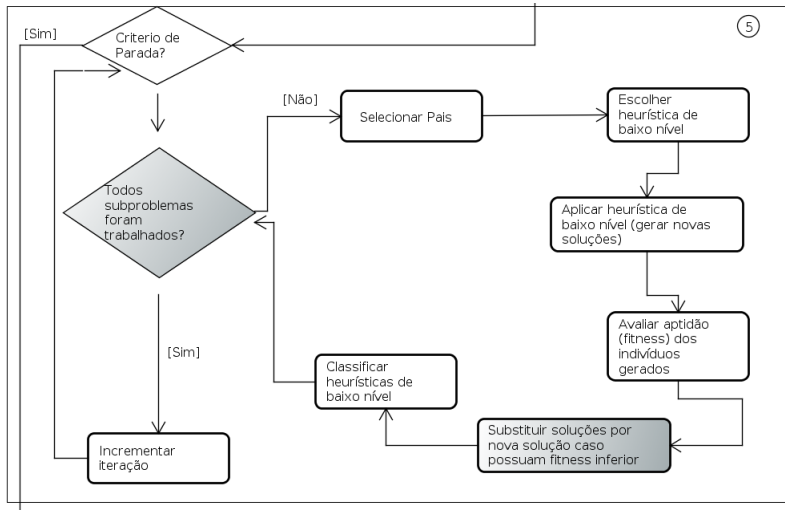
Hyper-heuristic for the Integration and Test Order Problem using Decomposition Approach (HITO-DA)

- Incorpora as características da HITO:
 - É hiper-heurística de aprendizado online para a seleção de LLHs perturbativas (operadores de mutação e cruzamento);
 - Busca selecionar a melhor LLH durante o processo de busca para o problema ITO.
- Trabalha com algoritmos de decomposição.

HITO-DA - Fluxo



HITO-DA - Fluxo



- Objetivos do problema
 - Número de operações referenciadas entre módulos
 - Número de atributos referenciadas entre módulos
- Meta-heurística
 - MOEA/D-DRA
- Algoritmos de seleção
 - FRRMAB
 - FRRCF
- Métricas de avaliação
 - Tchebycheff como função de *fitness* para o FIR do FRRMAB
 - Tchebycheff como função de *fitness* para o FIR do FRRCF

Fitness Rate Rank with Choice Function (FRRCF)

- Incorpora cálculos de FRR do FRRMAB;
- Usa a função de seleção da CF ao invés da função de seleção do MAB;
- Usa valores de FIR adaptados, pois na *Choice Function* não é empregado o conceito de janela de tempo (Equação 8).

$$FIR_{op} = FIR_{op} + \left\{ \begin{array}{l} 1 \text{ se } \frac{(F(p)-F(c))}{F(p)} > 0 \\ -1 \text{ se } \frac{(F(p)-F(c))}{F(p)} < 0 \\ 0 \text{ outro caso} \end{array} \right\} \quad (8)$$

FRRCF - Credit Assignment

Algoritmo 3: pseudocódigo da atribuição de crédito do FRRCF

```
1 Entrada: FIR, D, K
2 Saída: FRR
3 início
4   Classificar vetor de FIR em ordem decrescente;
5   para  $op \leftarrow 1$  até  $K$  faça
6      $Rank_{op} = FIR.getOpPosition(op)$ ;
7   fim
8   para  $op \leftarrow 1$  até  $K$  faça
9      $Decay_{op} = D^{Rank_{op}} * FIR_{op}$ ;
10  fim
11   $DecaySum = \sum_{op=1}^K Decay_{op}$ ;
12  para  $op \leftarrow 1$  até  $K$  faça
13     $FRR_{op} = Decay_{op} / DecaySum$ ;
14  fim
15  retorna FRR;
16 fim
```

Algoritmo 4: Pseudocódigo do método de seleção da FRRCF

```
1 Entrada:  $FRR$ ,  $WaitingTime_i$ ,  $\alpha$ ,  $\beta$ ,  $K$ 
2 Saída:  $op$ 
3 início
4   se Existem LLHs ainda não selecionados então
5      $op = \text{Selecionar aleatoriamente algum LLH ainda não selecionado};$ 
6   fim
7   senão
8      $op = \text{argmax}_{i=1\dots K}(FRR_i * \alpha + WaitingTime_i * \beta);$ 
9     para  $i \leftarrow 1$  até  $K$  faça
10       $WaitingTime_i = WaitingTime_i + 1$ 
11    fim
12     $WaitingTime_{op} = 0$ 
13  fim
14  retorna  $op$ ;
15 fim
```

- **RQ1:** Como são os resultados da HITO-DA com relação ao algoritmo MOEA/D-DRA?
- **RQ2:** Como são os resultados da HITO-DA com relação à hiper-heurística HITO?

- Meta-heurísticas (Instâncias da MOCAITO)
 - NSGA-II
 - SPEA2
 - MOEA/D-DRA (MOEA/D)

- HITO
 - Choice Function e NSGA-II (NSGA-II-CF)
 - MAB e NSGA-II (NSGA-II-MAB)
 - Choice Function e SPEA2 (SPEA2-CF)
 - MAB e SPEA2 (SPEA2-MAB)

- HITO-DA
 - FRRMAB e MOEA/D-DRA (MOEAD-MAB)
 - FRRCF e MOEA/D-DRA (MOEAD-CF)

Sistemas Utilizados

Sistema	Tipo	Versão	LOC	Classes	Aspectos
BCEL	OO	5.0	2999	45	-
JBoss	OO	6.0.0M5	8434	148	-
JHotDraw	OO	7.5.1	20273	197	-
MyBatis	OO	3.0.2.2	23535	331	-
AJHotDraw	OA	0.4	18586	288	31
AJHsqldb	OA	18	68550	275	30
Health-Watcher	OA	9	5479	95	22

Métrica de Qualidade Utilizada para HITO-DA

- Para FRRMAB:

$$n = \frac{g(x^j|\lambda^j, z^*) - g(y|\lambda^j, z^*)}{g(x^j|\lambda^j, z^*)} \quad (9)$$
$$FIR_{op} = FIR_{op} + n, n > 0$$

- Para FRRCF:

$$n = \frac{g(x^j|\lambda^j, z^*) - g(y|\lambda^j, z^*)}{g(x^j|\lambda^j, z^*)} \quad (10)$$
$$FIR_{op} = FIR_{op} + \left\{ \begin{array}{l} 1 \text{ se } n > 0 \\ -1 \text{ se } n < 0 \\ 0 \text{ outro caso} \end{array} \right\}$$

onde g é a função *Tchebycheff*

$g(\text{subproblema}|\text{peso associado, pontos ideais})$

Configuração de experimentos

	MOEAD-MAB	MOEAD-CF	NSGA-II-CF	NSGA-II-MAB	SPEA2-II-CF	SPEA2-II-MAB	MOEA/D
Pop/Arq	300	300	300	300	300	300	300
T	30	30	-	-	-	-	30
Nr	3, 6, 30	3, 6, 30	-	-	-	-	3, 6, 30
δ	0.9	0.9	-	-	-	-	0.9
D	0.1, 0.3, 0.5, 0.7, 0.9, 1.0	0.1, 0.3, 0.5, 0.7, 0.9, 1.0	-	-	-	-	-
W	150, 300	-	-	150, 300	-	150, 300	-
C	0.5, 1.0, 2.0, 5.0	-	-	0.5, 1.0, 2.0, 5.0	-	0.5, 1.0, 2.0, 5.0	-
α	-	1	1	-	1	-	-
β	-	0.10 0.05 0.01 0.005 0.001 0.00005	0.10 0.05 0.01 0.005 0.001 0.00005	-	0.10 0.05 0.01 0.005 0.001 0.00005	-	-

Configurações selecionadas

Alg	Pop/Arq	T. Muta	T. Cruza	T	Nr	δ	D	α	β	C	W
MOEAD-MAB	300	1	1	30	6	0.9	0.1	-	-	5.0	150
MOEAD-CF	300	1	1	30	6	0.9	0.3	1	0.10	-	-
NSGA-II-CF	300	1	1	-	-	-	-	1	0.00005	-	-
NSGA-II-MAB	300	1	1	-	-	-	-	-	-	2.0	150
SPEA2-CF	300	1	1	-	-	-	-	1	0.00005	-	-
SPEA2-MAB	300	1	1	-	-	-	-	-	-	0.5	150
MOEA/D	300	1	1	30	3	0.9	-	-	-	-	-
NSGA-II	300	0.02	0.95	-	-	-	-	-	-	-	-
SPEA2	300	0.02	0.95	-	-	-	-	-	-	-	-

- PF_{approx} contém o conjunto de soluções não dominadas de uma execução;
- PF_{known} é a união de todos conjuntos PF_{approx} de uma determinada instância, mas removendo soluções não dominadas;
- $PF_{trueknown}$ é a união de todos conjuntos PF_{known} , mas removendo soluções não dominadas.

Resultados da comparação da HITO-DA com o MOEA/D

Sistema	MOEA/D		MOEAD-CF		MOEAD-MAB	
	H_v	N_d	H_v	N_d	H_v	N_d
MyBatis	0,51604 (0,07289)	36,27	0,70966 (0,04470)	38,80	0,71292 (0,04501)	37,80
BCEL	0,66306 (0,04655)	28,33	0,72914 (0,00817)	29,13	0,73127 (0,00517)	29,17
JHotDraw	0,42821 (0,24306)	1,53	0,58827 (0,20472)	1,57	0,60494 (0,23102)	1,67
JBoss	0,70264 (0,46199)	1,00	1,00000 (0,00000)	1,00	1,00000 (0,00000)	1,00
AJHsqldb	0,26847 (0,10769)	11,57	0,74872 (0,07126)	19,60	0,78351 (0,06697)	22,80
AJHotDraw	0,27691 (0,14578)	2,60	0,64652 (0,10937)	5,63	0,65990 (0,16751)	5,27
Health-Watcher	0,80006 (0,31857)	1,03	1,00000 (0,00000)	1,00	0,99529 (0,02582)	1,00

HITO-DA, HITO, NSGA-II e SPEA2 em sistemas OO

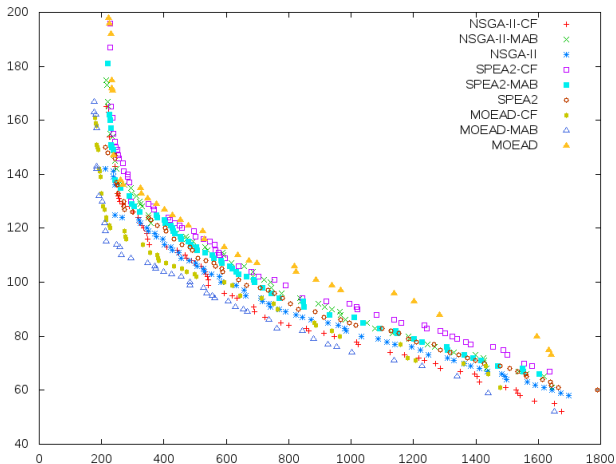
Alg	MyBatis		BCEL		JHotDraw		JBoss	
	H_v	N_d	H_v	N_d	H_v	N_d	H_v	N_d
NSGA-II	0,74316 (0,04679)	58,87	0,66824 (0,01069)	28,67	0,67662 (0,20095)	1,37	0,25498 (0,00000)	1,00
SPEA2	0,72504 (0,05090)	54,93	0,66919 (0,01121)	28,83	0,58732 (0,23660)	1,40	0,25498 (0,00000)	1,00
NSGA-II- CF	0,77862 (0,02563)	59,40	0,67451 (0,00291)	29,00	0,75597 (0,17912)	1,77	0,25498 (0,00000)	1,00
NSGA-II- MAB	0,74246 (0,05007)	57,80	0,67264 (0,00470)	29,00	0,75437 (0,19636)	1,77	0,25498 (0,00000)	1,00
SPEA2-CF	0,72687 (0,02653)	57,00	0,67364 (0,00488)	29,07	0,69985 (0,19621)	1,63	0,25498 (0,00000)	1,00
SPEA2- MAB	0,73881 (0,03911)	56,57	0,66782 (0,00917)	28,80	0,60077 (0,22012)	1,87	0,25498 (0,00000)	1,00
MOEAD- CF	0,78657 (0,03590)	38,80	0,66831 (0,00965)	29,13	0,59486 (0,20803)	1,57	0,25498 (0,00000)	1,00
MOEAD- MAB	0,78928 (0,03579)	37,80	0,67028 (0,00634)	29,17	0,60982 (0,23393)	1,67	0,25498 (0,00000)	1,00

HITO-DA, HITO, NSGA-II e SPEA2 em sistemas OA

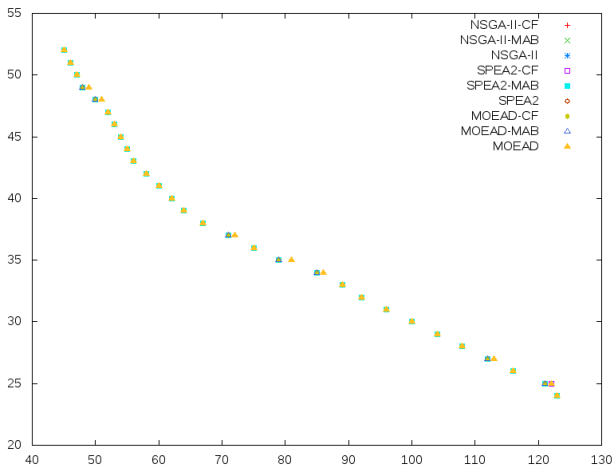
Alg	AJHsqldb		AJHotDraw		Health-Watcher	
	H_v	N_d	H_v	N_d	H_v	N_d
NSGA-II	0,60194 (0,09248)	30,23	0,62826 (0,09119)	4,10	0,50495 (0,00000)	1,00
SPEA2	0,57628 (0,08652)	28,13	0,54271 (0,11378)	4,73	0,50495 (0,00000)	1,00
NSGA-II-CF	0,69312 (0,10044)	32,40	0,76708 (0,09000)	5,17	0,50495 (0,00000)	1,00
NSGA-II-MAB	0,66027 (0,08791)	32,93	0,71146 (0,16133)	4,87	0,50495 (0,00000)	1,00
SPEA2-CF	0,60338 (0,05906)	28,27	0,73146 (0,10801)	4,33	0,50495 (0,00000)	1,00
SPEA2-MAB	0,61811 (0,08144)	25,07	0,69768 (0,12865)	5,40	0,50495 (0,00000)	1,00
MOEAD-CF	0,73957 (0,07613)	19,60	0,72859 (0,08777)	5,63	0,50495 (0,00000)	1,00
MOEAD-MAB	0,78730 (0,06985)	22,80	0,73937 (0,13117)	5,27	0,48829 (0,09128)	1,00

MyBatis

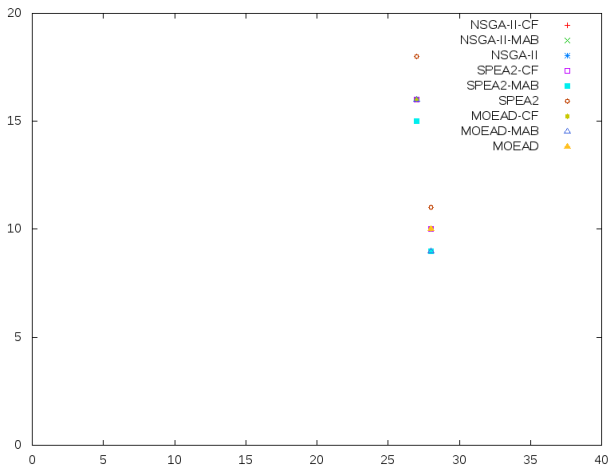
NSGA-II	SPEA2	NSGA-II-CF	NSGA-II-MAB	SPEA2-CF	SPEA2-MAB	MOEAD-CF	MOEAD-MAB
0	0	5	0	0	0	3	42



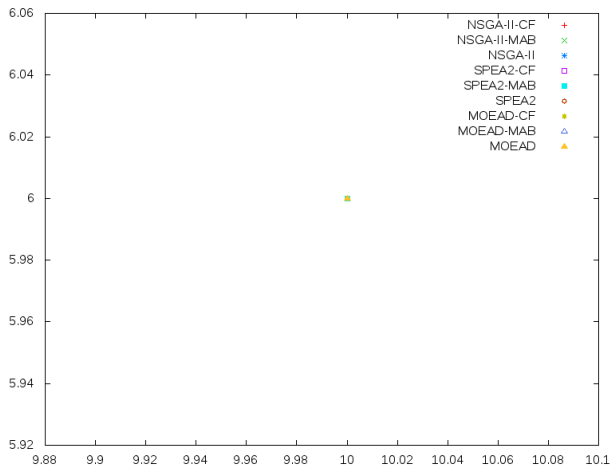
NSGA-II	SPEA2	NSGA-II-CF	NSGA-II-MAB	SPEA2-CF	SPEA2-MAB	MOEAD-CF	MOEAD-MAB
29	29	29	29	28	29	29	29



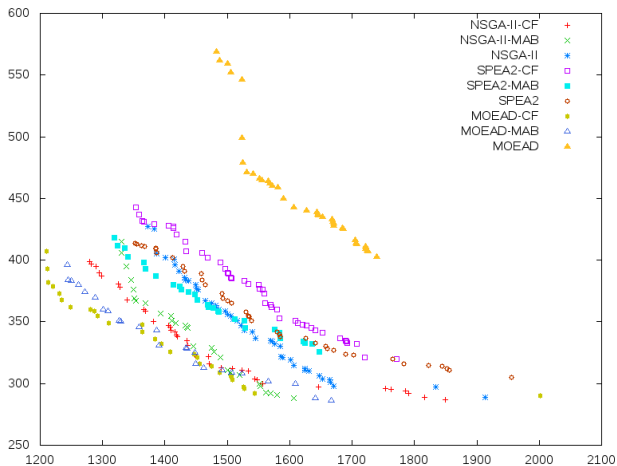
NSGA-II	SPEA2	NSGA-II-CF	NSGA-II-MAB	SPEA2-CF	SPEA2-MAB	MOEAD-CF	MOEAD-MAB
0	0	2	0	0	2	0	1



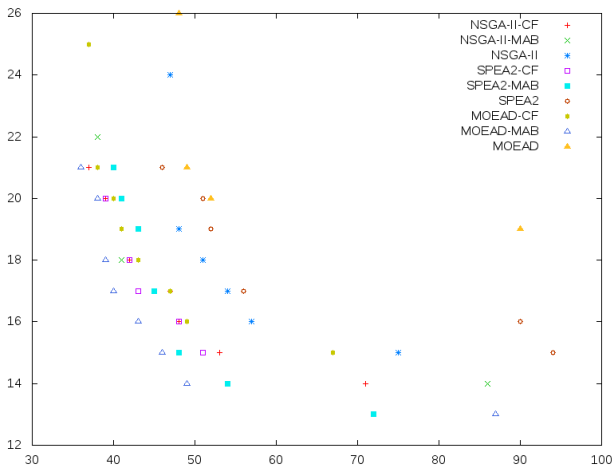
NSGA-II	SPEA2	NSGA-II-CF	NSGA-II-MAB	SPEA2-CF	SPEA2-MAB	MOEAD-CF	MOEAD-MAB
1	1	1	1	1	1	1	1



NSGA-II	SPEA2	NSGA-II-CF	NSGA-II-MAB	SPEA2-CF	SPEA2-MAB	MOEAD-CF	MOEAD-MAB
0	0	1	2	0	0	22	5

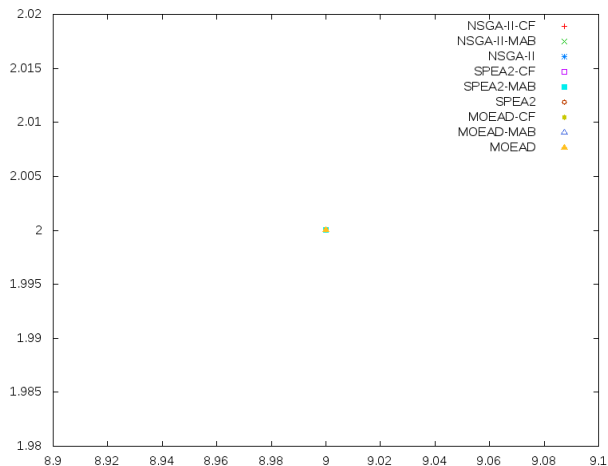


NSGA-II	SPEA2	NSGA-II-CF	NSGA-II-MAB	SPEA2-CF	SPEA2-MAB	MOEAD-CF	MOEAD-MAB
0	0	0	0	1	1	0	7



Health-Watcher

NSGA-II	SPEA2	NSGA-II-CF	NSGA-II-MAB	SPEA2-CF	SPEA2-MAB	MOEAD-CF	MOEAD-MAB
1	1	1	1	1	1	1	1



Média de escolhas de LLHs pela HITO-DA

Alg	Sistema	h1	h2	h3	h4	h5	h6	h7	h8	h9
MOEAD-CF	MyBatis	6599	6589	6592	6600	6621	6592	6906	6601	6595
	AJHsqldb	6578	6579	6578	6595	6857	6649	6616	6638	6604
	AJHotDraw	6620	6620	6619	6637	6656	6642	6625	6647	6631
	BCEL	6628	6633	6626	6640	6641	6641	6588	6670	6628
	JHotDraw	6607	6634	6633	6633	6646	6640	6597	6658	6647
	Health Watcher	6443	6662	6663	6649	6664	6671	6588	6673	6684
	JBoss	6632	6675	6672	6689	6806	6704	6157	6683	6677
MOEAD-MAB	MyBatis	7483	6695	7194	6393	6377	6426	6332	6343	6452
	AJHsqldb	7518	6686	7035	6338	6334	6338	6740	6340	6366
	AJHotDraw	7870	6445	6927	6334	6328	6329	6776	6336	6352
	BCEL	6848	6384	6547	6423	6331	6332	8140	6335	6356
	JHotDraw	7621	6477	6875	6346	6337	6338	7003	6347	6352
	Health Watcher	7724	6519	6937	6496	6333	6339	6637	6334	6376
	JBoss	7380	6405	6772	6333	6328	6331	7475	6333	6340

Média de classificação das instâncias (Friedman)

Instâncias	Classificação
NSGA-II-CF	2.5
MOEAD-MAB	3.642857142857143
NSGA-II-MAB	3.9285714285714284
MOEAD-CF	4.214285714285714
SPEA2-CF	4.214285714285714
NSGA-II	5.357142857142857
SPEA2-MAB	5.642857142857142
SPEA2	6.499999999999999
MOEA/D	9.0

Conclusões

- O MOEAD-MAB geralmente obteve melhores resultados do que a MOEAD-CF;
- O desempenho do MOEAD-CF no sistema *AJHsql/db* ainda motiva sua aplicação em sistemas maiores e mais complexos;
- A HITO-DA obteve melhores resultados do que a meta-heurística MOEA/D em quase todos os casos;
- A HITO-DA possui melhor desempenho em sistemas maiores, a HITO possui um melhor desempenho geral quando considerados sistemas menores.

Trabalhos Futuros

- Experimentos com sistemas maiores e mais complexos;
- Instanciar a HITO-DA com outros algoritmos de seleção;
- Instanciar a HITO-DA com outras meta-heurísticas baseadas em decomposição;
- Estudar o comportamento da HITO-DA para se resolver o problema ITO com um número maior de objetivos;
- Adaptar a HITO-DA para trabalhar com o problema ITO na presença de restrições de modularização;
- Criar uma variação da HITO-DA para outros problemas da SBSE, tais como a seleção ou priorização de casos de teste.

Muito Obrigado