O Impacto da Modelação na Resolução de Problemas de Satisfação Proposicional

Ruben Martins

Instituto Superior Técnico
Universidade Técnica de Lisboa

Mestrado em Matemática e Aplicações

Motivação

- Progresso notável na área de Satisfação Proposicional (SAT);
- Constante evolução das ferramentas de SAT.

Motivação

- Progresso notável na área de Satisfação Proposicional (SAT);
- Constante evolução das ferramentas de SAT.

- A modelação tem um elevado impacto na resolução de problemas de SAT.
- Necessidade de aprofundar a modelação de problemas:
  - *Social Golfer*, *Round Robin*, Completação de *Quasigroups*, Torres de Hanói

Resumo

- Conceitos Gerais
- *Social Golfer*
- *Round Robin*
- Completação de *Quasigroups*
- Torres de Hanói
- Conclusões e Trabalho Futuro

Resumo

- Conceitos Gerais
- *Social Golfer*
- *Round Robin*
- Completação de *Quasigroups*
- Torres de Hanói
- Conclusões e Trabalho Futuro

O que é SAT?

Definição (Problema de Satisfação Booleana (SAT))

O problema de Satisfação Booleana (SAT) consiste em, dada uma fórmula φ, decidir se existe uma atribuição ρ tal que $\rho \models \varphi$.

O que é SAT?

Definição (Problema de Satisfação Booleana (SAT))

O problema de Satisfação Booleana (SAT) consiste em, dada uma fórmula φ, decidir se existe uma atribuição ρ tal que $\rho \models \varphi$.

- $\varphi = (x_1 \vee \neg x_2) \wedge (x_2 \vee x_3) \wedge (\neg x_2 \vee \neg x_3)$

O que é SAT?

Definição (Problema de Satisfação Booleana (SAT))

O problema de Satisfação Booleana (SAT) consiste em, dada uma fórmula φ, decidir se existe uma atribuição ρ tal que $\rho \models \varphi$.

- $\varphi = (x_1 \vee \neg x_2) \wedge (x_2 \vee x_3) \wedge (\neg x_2 \vee \neg x_3)$
  - $\rho(x_1) = 1$
  - $\rho(x_2) = 1$
  - $\rho(x_3) = 0$

Porque é que SAT é importante?

- Importância Teórica:
  - Primeiro problema a ter sido demonstrado ser $\mathcal{NP}$-Completo (Cook'71).
- Importância Prática:
  - Resolução de problemas de planeamento,
  - Verificação de modelos,
  - Criptografia,
  - Bioinformática,
  - ...

Conceitos Gerais

- **Regra da Cláusula Unitária:**
  Dada uma cláusula com apenas um literal livre, temos que este literal terá que tomar o valor 1 para a cláusula ser satisfeita
  - $(x_1 \vee \neg x_2 \vee \neg x_3)$, x_3 terá que tomar o valor 0

- Regra da Cláusula Unitária:
  Dada uma cláusula com apenas um literal livre, temos que este literal terá que tomar o valor 1 para a cláusula ser satisfeita
  - $(x_1 \vee \neg x_2 \vee \neg x_3)$, x_3 terá que tomar o valor 0
- Propagação Unitária:
  Aplicação iterativa da Regra da Cláusula Unitária

- **Regra da Cláusula Unitária:**
  Dada uma cláusula com apenas um literal livre, temos que este literal terá que tomar o valor 1 para a cláusula ser satisfeita
  - $(x_1 \vee \neg x_2 \vee \neg x_3)$, x_3 terá que tomar o valor 0
- **Propagação Unitária:**
  Aplicação iterativa da Regra da Cláusula Unitária
- **Resolução:**
  $(x \vee \alpha) \wedge (\neg x \vee \beta) \vdash (\alpha \vee \beta)$

Algoritmos de SAT

- Em 1960, M. Davis e H. Putnam propuseram o algoritmo DP:
  - Uso da resolução para eliminar uma variável de cada vez;
  - Aplicação da Propagação Unitária.
- Em 1962, M. Davis, G. Logemann e D. Loveland proposeram um algoritmo alternativo (DPLL):
  - Invés de eliminar variáveis através da resolução, o algoritmo usa uma árvore binária de procura;
  - Aplicação da Propagação Unitária.

Algoritmo de SAT – DPLL

- Pesquisa em árvore
- Em cada passo:
  - `decidir()` Escolhe a variável a ser atríbuida
  - `deduzir()` Aplica a propagação unitária
  - `diagnosticar()` Se um conflito for identificado, então retroceder
    - Se não for possível retroceder, devolver UNSAT
    - Caso contrário, proceder com a propagação unitária
  - Se uma fórmula for satisfeita, devolver SAT
  - Caso contrário, proceder com outra decisão

Resumo

- Conceitos Gerais
- *Social Golfer*
- *Round Robin*
- Completação de *Quasigroups*
- Torres de Hanói
- Conclusões e Trabalho Futuro

Social Golfer

Definição (*Social Golfer*)

- O clube de golfe tem 32 membros;
- Cada membro joga uma vez por semana;
- Os membros jogam sempre em grupos 4;
- Cada dois jogadores jogam no máximo uma vez juntos.

Definição (*Social Golfer* – $< w, p, g >$)

- O calendário é composto por $p \times g$ jogadores, w semanas e g grupos por semana;
- Cada jogador joga golfe uma vez por semana;
- Cada jogador joga em grupos de p jogadores;
- Cada par de jogadores jogam no máximo uma vez juntos.

Social Golfer (Generalização)

Definição (*Social Golfer* – $< w, p, g >$)

- O calendário é composto por $p \times g$ jogadores, w semanas e g grupos por semana;
- Cada jogador joga golfe uma vez por semana;
- Cada jogador joga em grupos de p jogadores;
- Cada par de jogadores jogam no máximo uma vez juntos.

Semana	Grupo 1	Grupo 2
1	[1,2]	[3,4]
2	[1,3]	[2,4]
3	[1,4]	[2,3]

Simetrias (*Social Golfer*)

Semana	Grupo 1	Grupo 2
1	[1,2]	[3,4]
2	[1,3]	[2,4]
3	[1,4]	[2,3]

- Os jogadores são permutáveis;
- Jogadores no mesmo grupo são permutáveis;
- Grupos na mesma semana são permutáveis;
- As semanas são permutáveis.

Gent e Lynce VS. Martins

- Ambas as codificações usam restrições adicionais para quebrar simetrias.
- Gent e Lynce:
  As variáveis descrevem a posição em que cada jogador joga.
- Martins:
  As variáveis descrevem a semana em que cada grupo de jogadores joga.

Resultados Experimentais

- Comparação entre as diferentes codificações usando o `MiniSat`.

	Gent & Lynce			Martins		
Instância	#vars	#cls	tempo (s)	#vars	#cls	tempo (s)
$< 15, 2, 8 >$	12,103	78,547	1.19	1,800	38,040	0.1
$< 17, 2, 9 >$	31,671	219,267	333.2	2,601	62,730	0.2
$< 19, 2, 10 >$	47,500	336,700	293.38	3,610	97,850	0.5
$< 21, 2, 11 >$	68,607	495,979	3.46	4,851	145,992	1
$< 23, 2, 12 >$	96,048	706,032	97.18	6,348	210,036	1.57
$< 25, 2, 13 >$	130,975	976,651	138.45	8,125	293,150	14.39
$< 27, 2, 14 >$	174,636	1,318,492	597.43	10,206	398,790	6.85
$< 29, 2, 15 >$	228,375	1,743,075	946.56	12,615	530,700	9.81

Resultados Experimentais

- Comparação entre as diferentes codificações usando o `MiniSat`.

	Gent & Lynce			Martins		
Instância	#vars	#cls	tempo (s)	#vars	#cls	tempo (s)
$< 15, 2, 8 >$	12,103	78,547	1.19	1,800	38,040	0.1
$< 17, 2, 9 >$	31,671	219,267	333.2	2,601	62,730	0.2
$< 19, 2, 10 >$	47,500	336,700	293.38	3,610	97,850	0.5
$< 21, 2, 11 >$	68,607	495,979	3.46	4,851	145,992	1
$< 23, 2, 12 >$	96,048	706,032	97.18	6,348	210,036	1.57
$< 25, 2, 13 >$	130,975	976,651	138.45	8,125	293,150	14.39
$< 27, 2, 14 >$	174,636	1,318,492	597.43	10,206	398,790	6.85
$< 29, 2, 15 >$	228,375	1,743,075	946.56	12,615	530,700	9.81

- *Kirkman's Schoolgirl*:

	Gent & Lynce			Martins		
	#vars	#cls	tempo (s)	#vars	#cls	tempo (s)
$< 7, 3, 5 >$	5,775	42,555	226.12	3,185	726,285	12

Resumo

- Conceitos Gerais
- *Social Golfer*
- *Round Robin*
- Completação de *Quasigroups*
- Torres de Hanói
- Conclusões e Trabalho Futuro

Round Robin

Definição (*Round Robin*)

- Existem n (com n par) equipas e cada par de equipas joga entre si exactamente uma vez;
- A temporada dura n-1 semanas;
- Cada equipa joga um jogo em cada semana;
- Existem $\frac{n}{2}$ campos e em cada semana existe um jogo nesse campo;
- Nenhuma equipa joga mais do que duas vezes no mesmo campo durante o decorrer da temporada.

Definição (*Round Robin*)

- Existem n (com n par) equipas e cada par de equipas joga entre si exactamente uma vez;
- A temporada dura n-1 semanas;
- Cada equipa joga um jogo em cada semana;
- Existem $\frac{n}{2}$ campos e em cada semana existe um jogo nesse campo;
- Nenhuma equipa joga mais do que duas vezes no mesmo campo durante o decorrer da temporada.

	Sem 1	Sem 2	Sem 3	Sem 4	Sem 5
Campo 1	1 2	2 3	1 5	4 6	3 5
Campo 2	3 4	1 6	3 6	2 5	1 4
Campo 3	5 6	4 5	2 4	1 3	2 6

Simetrias (*Round Robin*)

	Sem 1	Sem 2	Sem 3	Sem 4	Sem 5
Campo 1	1 2	2 3	1 5	4 6	3 5
Campo 2	3 4	1 6	3 6	2 5	1 4
Campo 3	5 6	4 5	2 4	1 3	2 6

- Os jogadores são permutáveis;
- Jogadores no mesmo jogo são permutáveis;
- As semanas são permutáveis;
- Os campos são permutáveis.

Manyà e Béjar

	Sem 1	Sem 2	Sem 3	Sem 4	Sem 5
Campo 1	1 2	2 3	1 5	4 6	3 5
Campo 2	3 4	1 6	3 6	2 5	1 4
Campo 3	5 6	4 5	2 4	1 3	2 6

- As variáveis dividem-se em dois conjuntos:
  - p_{ij}^{1k}: descreve o jogador que joga na primeira posição de cada jogo;
  - p_{ij}^{2k}: descreve o jogador que joga na segunda posição de cada jogo.

Manyà e Béjar + < *simetrias* >

- Quebra das simetrias entre os campos:

	Sem 1	Sem 2	Sem 3	Sem 4	Sem 5
Campo 1	1 2	2 3	1 5	4 6	3 5
Campo 2	3 4	1 6	3 6	2 5	1 4
Campo 3	5 6	4 5	2 4	1 3	2 6

Manyà e Béjar + < *simetrias* >

- Quebra das simetrias entre os campos:

	Sem 1	Sem 2	Sem 3	Sem 4	Sem 5
Campo 1	1 2	2 3	1 5	4 6	3 5
Campo 2	3 4	1 6	3 6	2 5	1 4
Campo 3	5 6	4 5	2 4	1 3	2 6

- Quebra das simetrias entre as semanas:

	Sem 1	Sem 2	Sem 3	Sem 4	Sem 5
Campo 1	1 2	4 5	3 6	2 3	1 6
Campo 2	3 4	2 6	1 4	1 5	3 5
Campo 3	5 6	1 3	2 5	4 6	2 4

Manyà e Béjar + < *dummy* >

- Coluna *Dummy*:

	Sem 1	Sem 2	Sem 3	Sem 4	Sem 5	Dummy
Campo 1	1 2	4 5	3 6	2 3	1 6	4 5
Campo 2	3 4	2 6	1 4	1 5	3 5	2 6
Campo 3	5 6	1 3	2 5	4 6	2 4	1 3

	Sem 1	Sem 2	Sem 3	Sem 4	Sem 5
Campo 1	1 2	2 3	1 5	4 6	3 5
Campo 2	3 4	1 6	3 6	2 5	1 4
Campo 3	5 6	4 5	2 4	1 3	2 6

- As variáveis dividem-se em dois conjuntos:
  - $CAMPO_k(X)$: descreve o campo onde o grupo de jogadores X joga;
  - $SEM_j(X)$: descreve a semana onde o grupo de jogadores X joga.

	Sem 1	Sem 2	Sem 3	Sem 4	Sem 5
Campo 1	1 2	2 3	1 5	4 6	3 5
Campo 2	3 4	1 6	3 6	2 5	1 4
Campo 3	5 6	4 5	2 4	1 3	2 6

- As variáveis dividem-se em dois conjuntos:
  - $CAMPO_k(X)$: descreve o campo onde o grupo de jogadores X joga;
  - $SEM_j(X)$: descreve a semana onde o grupo de jogadores X joga.

  - Se X_1 e X_2 jogarem na mesma semana $\Rightarrow$ X_1 e X_2 jogam em campos diferentes;
  - Se X_1 e X_2 jogarem no mesmo campo $\Rightarrow$ X_1 e X_2 jogam em semanas diferentes.

Resultados Experimentais

- Codificação $< all >= \{< simetrias > \cup < dummy >\}$;
- Comparação entre as diferentes codificações usando o `Minisat` com um tempo limite de 6000 segundos.

n	Manyà & Béjar	<simetrias>	<dummy>	<all>	Martins
10	4.53	1.1	2.8	0.76	0.22
12	25.43	4.55	92	5	8.87
14	-	660.31	-	4493.65	4.4
16	-	-	-	-	-

Resumo

- Conceitos Gerais
- *Social Golfer*
- *Round Robin*
- Completação de *Quasigroups*
- Torres de Hanói
- Conclusões e Trabalho Futuro

Problema da Completação de *Quasigroups*

							1	
4								
	2							
				5		4		7
		8						
		1		9				
3			4					
	5		1					
			8		6			

Definição (Quasigroup)

Um **quasigroup** é um par ordenado $(Q, \cdot)$, onde Q é um conjunto e $\cdot$ é uma operação binária em Q, tal que, para cada a e b em Q, existem elementos únicos x e y em Q tal que: $a \cdot x = b$ e $y \cdot a = b$. A **ordem** n de um quasigroup é a cardinalidade do conjunto Q.

6	9	3	7	8	4	5	1	2
4	8	7	5	1	2	9	3	6
1	2	5	9	6	3	8	7	4
9	3	2	6	5	1	4	8	7
5	6	8	2	4	7	3	9	1
7	4	1	3	9	8	6	2	5
3	1	9	4	7	5	2	6	8
8	5	6	1	2	9	7	4	3
2	7	4	8	3	6	1	5	9

Definição (Quasigroup)

Um **quasigroup** é um par ordenado $(Q, \cdot)$, onde Q é um conjunto e $\cdot$ é uma operação binária em Q, tal que, para cada a e b em Q, existem elementos únicos x e y em Q tal que: $a \cdot x = b$ e $y \cdot a = b$. A **ordem** n de um quasigroup é a cardinalidade do conjunto Q.

Codificação SAT

							1	
4								
	2							
				5		4		7
		8						
		1		9				
3			4					
	5		1					
			8		6			

- Variáveis:
  - n variáveis Booleanas em cada célula,
  - A variável q_{xyz} é verdadeira se e só se o número z for atribuído à célula na linha x, coluna y.

6	9	3	7	8	4	5	1	2
4	8	7	5	1	2	9	3	6
1	2	5	9	6	3	8	7	4
9	3	2	6	5	1	4	8	7
5	6	8	2	4	7	3	9	1
7	4	1	3	9	8	6	2	5
3	1	9	4	7	5	2	6	8
8	5	6	1	2	9	7	4	3
2	7	4	8	3	6	1	5	9

Simetria Local – *lsym*$_{22}$ (2 linhas; 2 colunas)

6	9	3	7	8	4	5	1	2
4	8	7	5	1	2	9	3	6
1	2	5	9	6	3	8	7	4
9	3	2	6	5	1	4	8	7
5	6	8	2	4	7	3	9	1
7	4	1	3	9	8	6	2	5
3	1	9	4	7	5	2	6	8
8	5	6	1	2	9	7	4	3
2	7	4	8	3	6	1	5	9

Simetria Local – $lsym_{22}$ (2 linhas; 2 colunas)

6	9	3	7	8	4	5	1	2
4	8	7	5	1	2	9	3	6
1	2	5	9	6	3	8	7	4
9	3	2	6	5	1	4	8	7
5	6	8	2	4	7	3	9	1
7	4	1	3	9	8	6	2	5
3	1	9	4	2	5	7	6	8
8	5	6	1	7	9	2	4	3
2	7	4	8	3	6	1	5	9

Simetria Local – *lsym*$_{23}$ (2 linhas; 3 colunas)

6	9	3	7	8	4	5	1	2
4	8	7	5	1	2	9	3	6
1	2	5	9	6	3	8	7	4
9	3	2	6	5	1	4	8	7
5	6	8	2	4	7	3	9	1
7	4	1	3	9	8	6	2	5
3	1	9	4	7	5	2	6	8
8	5	6	1	2	9	7	4	3
2	7	4	8	3	6	1	5	9

Simetria Local – *lsym*$_{23}$ (2 linhas; 3 colunas)

6	9	3	7	8	4	5	1	2
4	8	7	5	1	2	9	3	6
1	2	5	9	6	3	8	7	4
9	4	2	3	5	1	6	8	7
5	6	8	2	4	7	3	9	1
7	3	1	6	9	8	4	2	5
3	1	9	4	7	5	2	6	8
8	5	6	1	2	9	7	4	3
2	7	4	8	3	6	1	5	9

Simetria Local – *lsym*$_{32}$ (3 linhas; 2 colunas)

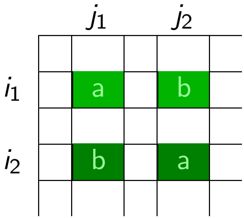
6	9	3	7	8	4	5	1	2
4	8	7	5	1	2	9	3	6
1	2	5	9	6	3	8	7	4
9	3	2	6	5	1	4	8	7
5	6	8	2	4	7	3	9	1
7	4	1	3	9	8	6	2	5
3	1	9	4	7	5	2	6	8
8	5	6	1	2	9	7	4	3
2	7	4	8	3	6	1	5	9

Simetria Local – $lsym_{32}$ (3 linhas; 2 colunas)

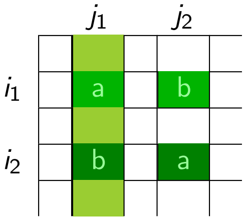
6	9	3	7	8	4	5	1	2
4	8	7	5	1	2	9	3	6
1	2	5	9	6	3	7	8	4
9	3	2	6	5	1	8	4	7
5	6	8	2	4	7	3	9	1
7	4	1	3	9	8	6	2	5
3	1	9	4	7	5	2	6	8
8	5	6	1	2	9	4	7	3
2	7	4	8	3	6	1	5	9

Quebra de Simetrias Locais – $lsym_{22}$

- $\mathcal{Q}[i_1, j_1] < \mathcal{Q}[i_2, j_1]$



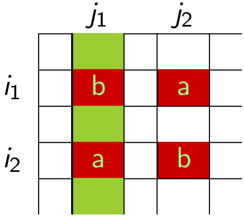
- $\mathcal{Q}[i_1, j_1] < \mathcal{Q}[i_2, j_1]$



- Impomos uma ordem lexicográfica no subconjunto de j_1:
  1. $\forall_{b>a}\ \neg(q_{i_1 j_1 b} \land q_{i_1 j_2 a} \land q_{i_2 j_1 a} \land q_{i_2 j_2 b})$
  2. $\forall_{a>b}\ \neg(q_{i_1 j_1 a} \land q_{i_1 j_2 b} \land q_{i_2 j_1 b} \land q_{i_2 j_2 a})$

Quebra de Simetrias Locais – $lsym_{22}$

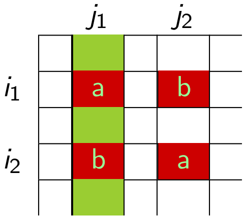
- $\mathcal{Q}[i_1, j_1] < \mathcal{Q}[i_2, j_1]$



- Impomos uma ordem lexicográfica no subconjunto de j_1:
  1. $\forall_{b>a}\ \neg(q_{i_1 j_1 b} \wedge q_{i_1 j_2 a} \wedge q_{i_2 j_1 a} \wedge q_{i_2 j_2 b})$
  2. $\forall_{a>b}\ \neg(q_{i_1 j_1 a} \wedge q_{i_1 j_2 b} \wedge q_{i_2 j_1 b} \wedge q_{i_2 j_2 a})$

Quebra de Simetrias Locais – $lsym_{22}$

- $\boxed{\mathcal{Q}[i_1, j_1] < \mathcal{Q}[i_2, j_1]}$

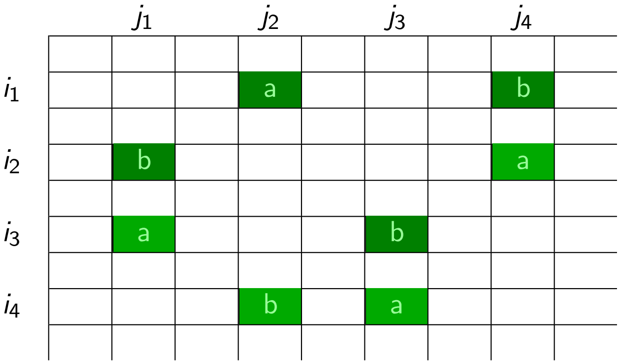


- Impomos uma ordem lexicográfica no subconjunto de j_1:
  1. $\forall_{b>a}\ \neg(q_{i_1 j_1 b} \wedge q_{i_1 j_2 a} \wedge q_{i_2 j_1 a} \wedge q_{i_2 j_2 b})$
  2. $\boxed{\forall_{a>b}\ \neg(q_{i_1 j_1 a} \wedge q_{i_1 j_2 b} \wedge q_{i_2 j_1 b} \wedge q_{i_2 j_2 a})}$

Outras Simetrias Locais

		j_1		j_2		j_3		j_4	
i_1				b				a	
i_2		a						b	
i_3		b				a			
i_4				a		b			

Outras Simetrias Locais



Resultados Experimentais (Instâncias com Solução)

- Redução (em percentagem) do número de soluções

$lsym_{22}$	$lsym_{23}$	$lsym_{32}$	$lsym_{all}$
77.191	8.910	10.934	81.668

Resultados Experimentais (Instâncias com Solução)

- Redução (em percentagem) do número de soluções

$lsym_{22}$	$lsym_{23}$	$lsym_{32}$	$lsym_{all}$
77.191	8.910	10.934	81.668

- Comparação entre as diferentes codificações usando o `Satz` com um tempo limite de 6000 segundos.

Ordem	w/o *lsym*		$lsym_{22}$		$lsym_{32}$		$lsym_{23}$	
35	100	0.66	100	0.64	100	0.825	100	0.635
37	100	3.44	100	3.37	100	3.495	100	4.015
40	100	18.76	100	18.63	100	26.645	100	19.92
43	90	120.66	91	134.41	90	156.11	89	170.655
45	68	665.22	69	633.55	68	802.835	70	740.22

Resultados Experimentais (blindSatz)

- `Satz`
  O algoritmo de decisão implementado no `Satz` escolhe a variável que quando atribuída irá implicar um maior número de atribuições através de propagação unitária.

Resultados Experimentais (blindSatz)

- `Satz`

  O algoritmo de decisão implementado no `Satz` escolhe a variável que quando atribuída irá implicar um maior número de atribuições através de propagação unitária.

- `blindSatz`

  Escolhemos a *primeira variável livre* para ramificar. Isto faz com que este algoritmo escolha as variáveis numa ordem fixa.

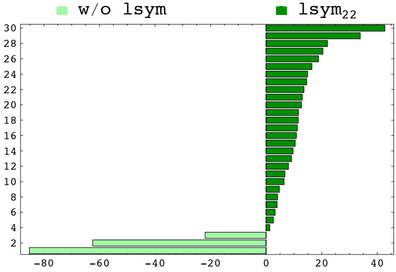
Resultados Experimentais (Instâncias com Solução)

- Comparação entre as diferentes codificações usando o blind`Satz` com um tempo limite de 1000 segundos.

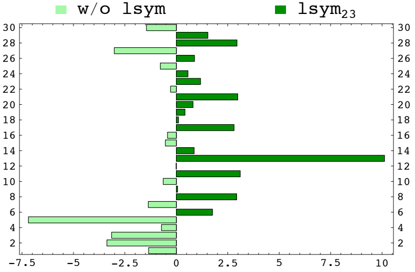
w/o $lsym$	$lsym_{22}$	$lsym_{32}$	$lsym_{23}$	$lsym_{all}$
88.97	81.57	88.87	88.97	81.095

- Comparação entre as diferentes codificações usando o blindSatz com um tempo limite de 1000 segundos.

w/o lsym	$lsym_{22}$	$lsym_{32}$	$lsym_{23}$	$lsym_{all}$
88.97	81.57	88.87	88.97	81.095



27 Instâncias (1.32;12.57;42.74)%

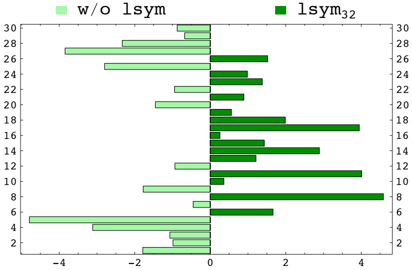


16 Instâncias (0.06;2.07;10.12)%

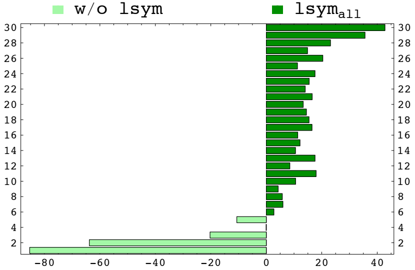
Resultados Experimentais (Instâncias com Solução)

- Comparação entre as diferentes codificações usando o blind`Satz` com um tempo limite de 1000 segundos.

w/o lsym	$lsym_{22}$	$lsym_{32}$	$lsym_{23}$	$lsym_{all}$
88.97	81.57	88.87	88.97	81.095



15 Instâncias (0.26;1.85;4.59)%



25 Instâncias (2.76;15.17;42.81)%

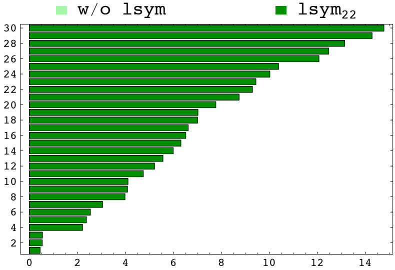
Resultados Experimentais (Instâncias sem Solução)

- Comparação entre as diferentes codificações usando o blind`Satz` com um tempo limite de 1000 segundos.

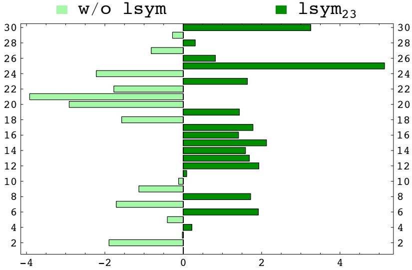
w/o lsym	$lsym_{22}$	$lsym_{32}$	$lsym_{23}$	$lsym_{all}$
376.075	360.655	378.52	377.955	358.47

- Comparação entre as diferentes codificações usando o blindSatz com um tempo limite de 1000 segundos.

w/o lsym	$lsym_{22}$	$lsym_{32}$	$lsym_{23}$	$lsym_{all}$
376.075	360.655	378.52	377.955	358.47



30 Instâncias (0.45;6.71;14.78)%

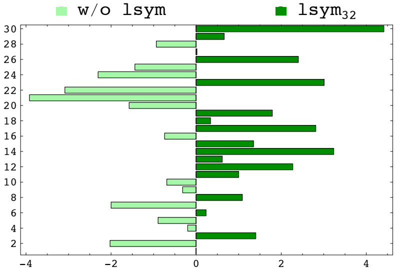


16 Instâncias (0.09;1.69;5.14)%

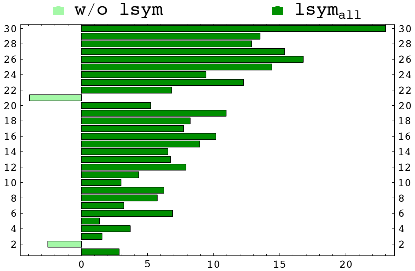
Resultados Experimentais (Instâncias sem Solução)

- Comparação entre as diferentes codificações usando o blindSatz com um tempo limite de 1000 segundos.

w/o lsym	$lsym_{22}$	$lsym_{32}$	$lsym_{23}$	$lsym_{all}$
376.075	360.655	378.52	377.955	358.47



16 Instâncias (0.02;1.67;4.41)%



28 Instâncias (1.37;8.42;23)%

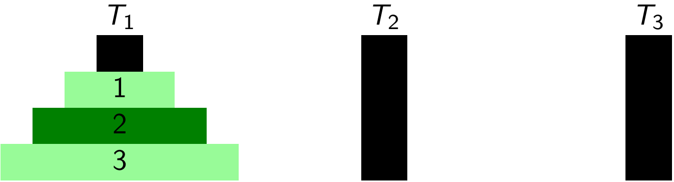
Resumo

- Conceitos Gerais
- *Social Golfer*
- *Round Robin*
- Completação de *Quasigroups*
- Torres de Hanói
- Conclusões e Trabalho Futuro

Torres de Hanói

Definição (Torres de Hanói)

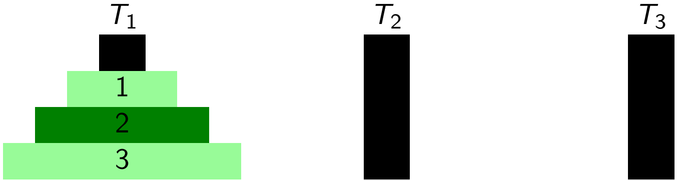
- 3 torres;
- n anéis de diferentes tamanhos;
- Situação Inicial: anéis empilhados em ordem crescente de tamanho na primeira torre;
- Objectivo: anéis empilhados em ordem crescente de tamanho na terceira torre.



Torres de Hanói

Regras (Torres de Hanói)

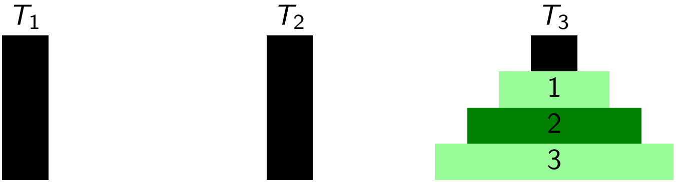
- Apenas um disco pode ser movido de cada vez;
- Cada movimento consiste em tirar o disco que se encontra no topo de uma das Torres e colocá-lo noutra Torre, no topo de outros discos que possam já estar presentes nessa Torre;
- Nenhum disco pode ser colocado em cima de um disco mais pequeno.



Torres de Hanói

Regras (Torres de Hanói)

- Apenas um disco pode ser movido de cada vez;
- Cada movimento consiste em tirar o disco que se encontra no topo de uma das Torres e colocá-lo noutra Torre, no topo de outros discos que possam já estar presentes nessa Torre;
- Nenhum disco pode ser colocado em cima de um disco mais pequeno.



Existe sempre uma solução com $2^n - 1$ passos

- Têm por base a linguagem STRIPS.
- Selman e Kautz: A codificação descreve o movimento de um disco para cima de outro disco ou de uma torre.
  - $move(d, dt, i) \equiv (obj(d, i) \wedge origem(dt, i) \wedge dest(dt, i))$
  - $on(d, dt, i)$; $clear(dt, i)$
- Prestwich: A codificação descreve o movimento de um disco para cima de uma torre.
  - $move(d, t, i) \equiv (obj(d, i) \wedge origem(t, i) \wedge dest(t, i))$
  - $on(d, t, i)$

Torres de Hanói em SAT (Martins)

Propriedades:

- **[seq]:** A sequência dos discos a serem movidos é dada por: $S_1 = \{1\};\ S_n = \{S_{n-1}, n, S_{n-1}\}$
- **[pp1]:** Dada um problema de Torres de Hanói com n discos, podemos obter a solução determinando apenas os primeiros $2^{n-1} - 1$ passos.
- **[pp2]:** Discos da mesma paridade nunca podem estar em directo contacto.
- **[pp3]:** Existe no máximo um disco par no topo das três torres.
- **[pp4]:** Se considerarmos um disco extra por torre, com a respectiva numeração de $n+1$, $n+2$ e $n+3$, podemos garantir que existe exactamente um disco par no topo das três torres.

- Comparação entre as diferentes codificações usando o `PicoSAT` com um tempo limite de 10000 segundos.

n	Selman	Prestwich	[seq]	[pp1]	[pp2]	[pp3]	[pp4]
4	0.16	0.01	0.01	0.01	0.01	0.02	0.02
5	8.31	0.14	0.01	0.01	0.08	0.10	0.15
6	54.70	1.07	0.05	0.13	1.10	1.94	1.36
7	5252.27	10.97	0.32	0.99	14.38	19.44	12.15
8	-	332.32	1.44	16.08	154.45	187.28	220.30
9	-	-	3.47	391.12	2616.45	-	3003.58
10	-	-	71.66	-	-	-	-
11	-	-	1345.33	-	-	-	-
12	-	-	6950.09	-	-	-	-

- {[seq],[pp1]} permite resolver o problema sem procura:

n	# vars	#claus	{[seq],[pp1]}
4	166	1,106	0.01
5	405	3,140	0.02
6	948	8,319	0.07
7	2,163	21,091	0.15
8	4,850	51,872	0.39
9	10,737	124,758	0.85
10	23,536	294,917	2.32
11	51,183	687,533	5.04
12	110,574	1,584,462	12.37

Codificação Simplificada

- Aproveitamos a propriedade [seq] para usarmos como variáveis apenas as variáveis $on(d, t, i)$;
- Utilizamos as propriedades [pp1] e [pp2].

Codificação Simplificada

- Aproveitamos a propriedade [seq] para usarmos como variáveis apenas as variáveis $on(d, t, i)$;
- Utilizamos as propriedades [pp1] e [pp2].

n	Codificação	#vars	#claus
4	<seq,pp1>	166	1,106
	<simp>	84	232
5	<seq,pp1>	405	3,140
	<simp>	225	711
6	<seq,pp1>	948	8,319
	<simp>	558	1,902
7	<seq,pp1>	2,163	21,091
	<simp>	1,323	4,911
8	<seq,pp1>	4,850	51,872
	<simp>	3,048	11,984

n	Codificação	#vars	#claus
9	<seq,pp1>	10,737	124,758
	<simp>	6,885	28,971
10	<seq,pp1>	23,536	294,917
	<simp>	15,330	67,846
11	<seq,pp1>	51,183	687,533
	<simp>	33,759	158,427
12	<seq,pp1>	110,574	1,584,462
	<simp>	73,692	362,160

Resumo

- Conceitos Gerais
- *Social Golfer*
- *Round Robin*
- Completação de *Quasigroups*
- Torres de Hanói
- Conclusões e Trabalho Futuro

Conclusões e Trabalho Futuro

- Modelação deve ser adequada a cada problema;
- Elevado impacto da modelação na resolução de problemas de SAT:
  - Torres de Hanói, reduzimos um problema inicialmente difícil a um problema que pode ser resolvido sem procura.
- Exploração de novas técnicas de modelação:
  - Identificação e quebra eficiente de Simetrias Locais.