

Métodos Heurísticos aplicados ao Problema de Tabela-Horário de Universidades

Edmar Hell Kampke
Maria Claudia Silva Boeres

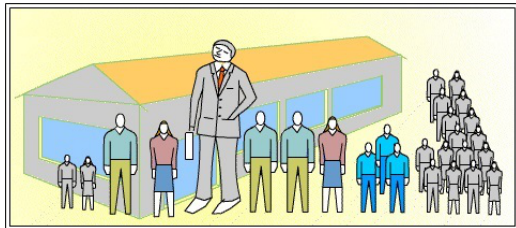
Laboratório de Otimização – LabOtim
Universidade Federal do Espírito Santo – UFES

12 de Setembro de 2023

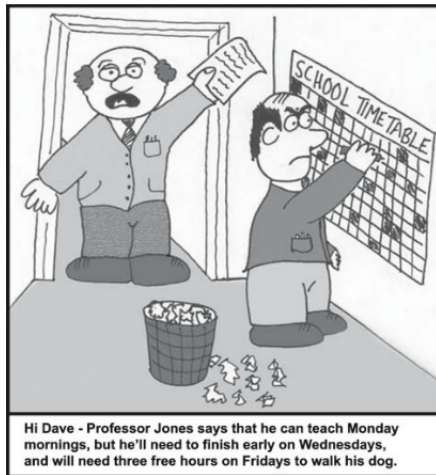


- ▶ Introdução
- ▶ Definição do Problema
- ▶ Revisão da Literatura
- ▶ Métodos
- ▶ Resultados Computacionais
 - Detalhes de Implementação
 - Escolha dos Parâmetros
 - Análise dos Resultados
- ▶ Conclusões e Trabalhos Futuros
- ▶ Referências

- ▶ Problemas de Agendamento (*Scheduling*)
- ▶ Problemas de Tabela-Horário (PTH)
- ▶ Algumas aplicações:
 - Escalas de Funcionários
 - Partidas de um Campeonato Esportivo
 - Tabela-horário Educacionais



- ▶ Problema de Tabela-Horário Educacionais
 - Escolas de Ensino Médio
 - **Universidades (PTHU)**
 - Exames



- ▶ NP-Completo para a maioria das formulações (Schaerf [1995])
- ▶ As formulações variam entre as universidades
- ▶ Dificuldade aumenta com o aumento no número de restrições
- ▶ Restrições Fracas e Fortes são definidas
- ▶ Restrições Fortes devem ser satisfeitas a qualquer custo
 - Qualquer violação inviabiliza a solução
- ▶ É desejável que Restrições Fracas sejam satisfeitas
 - O não atendimento não inviabiliza a solução

- ▶ *The International Series of Conferences on the Practice and Theory of Automated Timetabling* (PATAT)
- ▶ Organizam competições de tabela-horário educacional
- ▶ Objetivo
 - Gerar pesquisas com diferentes abordagens e preencher o espaço entre a pesquisa e prática nessa área de pesquisa.
- ▶ ITC – *International Timetabling Competition*
 - ITC-2002 (Universidades)
 - **ITC-2007 (Universidades – Três Formulações)**
 - * PTHU – Exames Finais
 - * PTHU – Matrícula dos Alunos
 - * **PTHU – Currículo de Cursos**
 - ITC-2011 (Escolas de Ensino Médio)

► ITC–2019 (Universidades)

Important Dates

- **August 30, 2018:** Competition is announced, **sample data instances** are published.
- **November 15, 2018:** **Early data instances** are published.
- **February 1, 2019:** The deadline for the first milestone submission of the results for early instances to announce the three best competitors at this point.
- **June 1, 2019:** The deadline for the second milestone submission of the results for early instances to announce the three best competitors at this point.
- **September 18, 2019:** Middle data instances are published.
- **November 8, 2019:** Late data instances are published.
- **November 18, 2019:** The final deadline for submission of the results for all the competition instances and the summary paper.
- **January 1, 2020:** The deadline for the publication of the solvers competing for the open source price.
- **January 15, 2020:** Finalists are published and winners are informed.
- **February–March, 2020:** Submission deadline for PATAT 2020 competition track where any paper related to the competition can be submitted.
- **August, 2020:** Winners are announced at PATAT 2020.
- **Autumn, 2020:** Submission deadline for PATAT 2020 journal special issue.

<https://www.itc2019.org>

- ▶ A modelagem do problema foi baseada nas instâncias utilizadas pelo ITC-2007.
- ▶ Cada instância possui as seguintes informações:
 - ▶ Dias, Horários e Períodos
 - ▶ Disciplinas e Professores
 - ▶ Salas
 - ▶ Currículo
 - ▶ Indisponibilidades

- ▶ T é uma solução (ou Tabela-Horário) para o problema
 - d é o número de dias de aula na semana em T ($d = 2$)
 - Cada dia é dividido em q horários ($q = 2$)
 - nr é o número de salas disponíveis ($nr = 2$)
- ▶ T representa a alocação de todas as aulas de todas as disciplinas a um timeslot t

| Sala | Dia 0 | | Dia 1 | |
|-------|-------|-------|-------|-------|
| | p_0 | p_1 | p_2 | p_3 |
| r_0 | | | | |
| r_1 | | t | | |

Name: Toy3
Courses: 3
Rooms: 2
Days: 2
Periods_per_day: 2
Curricula: 2
Constraints: 2

COURSES:
DiscMath Edmar 2 2 40
CompProg Geraldo 2 2 35
GraphThe Edmar 2 3 20

ROOMS:
rA 38
rB 32

CURRICULA:
Cur1 2 DiscMath CompProg
Cur2 1 GraphThe

UNAVAILABILITY _ CONSTRAINTS:
CompProg 0 0
CompProg 1 1
END.

► Restrições Fortes:

- **H1.** *Aulas*
- **H2.** *Ocupação de Salas*
- **H3.** *Conflitos–Professor e Currículo*
- **H4.** *Disponibilidade*

► Restrições Fracas:

- **S1.** *Capacidade de Sala*
- **S2.** *Estabilidade de Sala*
- **S3.** *Dias Mínimos*
- **S4.** *Aulas Isoladas*

Função Objetivo

Minimizar $f(s) = f_{Fortes}(s) + f_{Fracas}(s)$

Custo – Restrições Fortes e Fracas

- ▶ Se s é uma solução viável, então $f_{Fortes}(s) = 0$.
- ▶ $f_{Fracas}(s) = \sum_{i=1}^4 \alpha_i \cdot \omega_i$
- ▶ α_i e ω_i são os pesos e o número total de violações, respectivamente, de cada Restrição Fraca i .
- ▶ $\alpha_1 = 1$, $\alpha_2 = 1$, $\alpha_3 = 5$ e $\alpha_4 = 2$

- ▶ A maior parte dos trabalhos recentes usa métodos heurísticos para resolver o problema
 - Procedimentos inteligentes de busca local
 - Explora o espaço de soluções
 - Escapa de ótimos locais
- ▶ Müller [2009] foi o vencedor da terceira formulação do ITC-2007 utilizando *Conflict Based Statistics* para gerar solução inicial e *Hill Climbing*, *Great Deluge* e *Simulated Annealing* para refinamento da solução e com movimentos específicos.
- ▶ Lü e Hao [2010] usaram *Adaptive Tabu Search* para encontrar uma solução da terceira formulação do ITC-2007. Além disso propuseram um método guloso para gerar uma solução inicial. Foram o segundo colocado na competição.

- ▶ Lü et al. [2011] analisou quatro movimentos clássicos da busca local para o problema
- ▶ Os movimentos foram estudados separadamente e combinados
- ▶ Três critérios foram considerados na análise
 - Porcentagem de vizinhos melhores
 - A intensidade de melhoria
 - O número de etapas da busca
- ▶ O algoritmo *Steepest Descent* (SD) foi usado como técnica de busca local na análise

- ▶ Rocha [2013] utiliza GRASP com *Hill Climbing* e *Simulated Annealing* como busca local para solução do PTHU da terceira formulação do ITC-2007.
- ▶ Segatto et al. [2015] deu continuidade ao trabalho de Rocha [2013], aplicando Cadeia de Kempe como um novo movimento para a geração de vizinhos na busca local.
- ▶ Kampke et al. [2017] estenderam a análise de Lü et al. [2011] adicionando movimentos específicos propostos por Müller [2009].
- ▶ Kampke et al. [2019] apresentaram um novo método para construção de uma solução inicial, usando fluxo em redes.

Algoritmo 1: Pseudocódigo genérico do GRASP.

Entrada: $MaxIter, \alpha$

Saída: Solução S^*

```
1  $f^* \leftarrow \infty$ ;  
2 para  $i \leftarrow 1$  até  $MaxIter$  faça  
3    $S \leftarrow \text{ConstrucaoSolucaoInicial}(\alpha)$ ;  
4    $S \leftarrow \text{BuscaLocal}(S)$ ;  
5   se  $f(S) < f^*$  então  
6      $S^* \leftarrow S$ ;  
7      $f^* \leftarrow f(S)$ ;  
8   fim  
9 fim
```

- ▶ Lista de aulas não alocadas é ordenada pela quantidade de períodos disponíveis (que não causam violação de restrição forte).
- ▶ É possível alocar a primeira aula da lista em algum período?
 - ▶ Se não houver horário disponível: Explosão
 - ▶ Retira-se aleatoriamente uma aula já alocada para abrir espaço na tabela-horário
- ▶ Lista de Candidatos (LC): Timeslots disponíveis para a aula ser alocada.

- ▶ Lista Restrita de Candidatos:
 - ▶ LC é ordenada pelo custo de alocação da aula em cada timeslot
 - ▶ Escolhe-se aleatoriamente um timeslot dentre os $\alpha\%$ elementos de LC.
- ▶ A construção termina quando todas aulas estão inseridas na tabela-horário.
- ▶ O procedimento sempre retorna uma solução viável

Name: Toy3

Courses: 3

Rooms: 2

Days: 2

Periods_per_day: 2

Curricula: 2

Constraints: 2

COURSES:

DiscMath Edmar 2 2 40

CompProg Geraldo 2 2 35

GraphThe Edmar 2 3 20

ROOMS:

rA 38

rB 32

CURRICULA:

Cur1 2 DiscMath CompProg

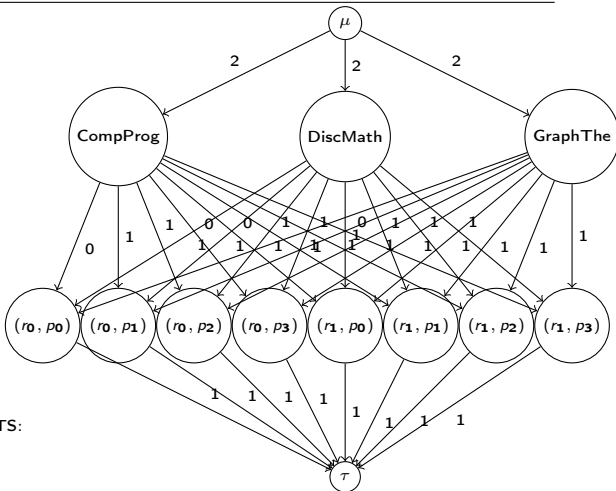
Cur2 1 GraphThe

UNAVAILABILITY_CONSTRAINTS:

CompProg 0 0

CompProg 1 1

END.



- ▶ Realizada de forma gulosa para produzir soluções de melhor qualidade.
- ▶ Cada iteração é composta por duas operações distintas: escolha da aula e escolha do timeslot.
- ▶ As escolhas são realizadas considerando algumas definições sobre a tabela-horário.

- ▶ $apd_i(S)$: Total de períodos disponíveis para uma disciplina i em S .
- ▶ $aps_i(S)$: Total de timeslots disponíveis para uma disciplina i em S .
- ▶ $nl_i(S)$: Total de aulas não alocadas da disciplina i em S .
- ▶ $uac_{i,j}(S)$: Total de aulas das disciplinas ainda não alocadas completamente que se tornam indisponíveis para alocação no período j após a inserção de uma aula da disciplina i no período j .
- ▶ $conf_i$: Número de disciplinas que compartilham estudantes ou professores com a disciplina i .

- ▶ A escolha da aula é realizada de acordo com os seguintes critérios:

1. Escolhe-se a disciplina com o menor valor de $\frac{apd_i(S)}{\sqrt{nl_i(S)}}$.
2. Se existir múltiplas disciplinas com o mesmo valor, é escolhida a disciplina com o menor valor de $aps_i(S)\sqrt{nl_i(S)}$.
3. Se ainda houver a ocorrência de mais de uma disciplina com o menor valor, é escolhida a disciplina com maior valor de $conf_i$.
4. Se ainda existir empates isso é resolvido seguindo a ordem alfabética do nome da disciplina.

- ▶ Em seguida, é escolhido o período j e sala k com o menor valor da função $g(j, k)$, dada pela fórmula:

$$g(j, k) = k_1 \times uac_{i,j}(S) + k_2 \times \Delta f_S(i, j, k) \quad (1)$$

- ▶ $\Delta f_S(i, j, k)$ o valor da penalidade das restrições fracas sobre a possível alocação da aula i no período j e sala k .
- ▶ Os pesos k_1 e k_2 são relativos às restrições fortes e fracas, respectivamente.

Algoritmo 2: Pseudocódigo genérico do SD.

Entrada: Solução Inicial S

Saída: Melhor Solução S^*

```
1  $S^* \leftarrow S$ ;  
2  $S_{atual} \leftarrow S$ ;  
3 enquanto  $S_{atual} = S^*$  faça  
4   Escolher  $m' \in M(S^*)$ , tal que  $f(S^* \otimes m') = \min_{m \in M(S)} f(S^* \otimes m)$  ;  
5    $S_{atual} \leftarrow f(S^* \otimes m')$ ;  
6   se  $f(S_{atual}) < f(S^*)$  então  
7      $S^* \leftarrow S_{atual}$ ;  
8   fim  
9 fim
```

Algoritmo 3: Pseudocódigo genérico do Hill Climbing

Entrada: Solução S , N , k

Saída: Melhor Solução S^*

```
1  $i \leftarrow 0$  ;  
2  $S^* \leftarrow S$  ;  
3 enquanto  $i < N$  faça  
4    $S' \leftarrow \text{GeraVizinho}(S^*, k)$  ;  
5    $\Delta f \leftarrow f(S') - f(S^*)$  ;  
6   se  $\Delta f < 0$  então  
7      $S^* \leftarrow S'$  ;  
8      $i \leftarrow 0$  ;  
9   fim  
10   $i \leftarrow i + 1$  ;  
11 fim
```

Algoritmo 4: Pseudocódigo genérico do Simulated Annealing

Entrada: Solução S , T_i , T_f , β , N_v

Saída: Solução S^*

```

1   $T \leftarrow T_i$  ;
2   $S^* \leftarrow S$  ;
3   $S_{atual} \leftarrow S$  ;
4  enquanto  $T > T_f$  faça
5      para  $i \leftarrow 1$  até  $N_v$  faça
6           $S' \leftarrow \text{GeraVizinho}(S_{atual})$  ;
7           $\Delta f \leftarrow f(S') - f(S^*)$  ;
8          se  $\Delta f < 0$  então
9               $S^* \leftarrow S'$  ;
10              $S_{atual} \leftarrow S'$  ;
11             fim
12         senão
13             Gere um número aleatório  $p \in (0, 1]$  ;
14             se  $p < e^{-\Delta f / T}$  então
15                  $S_{atual} \leftarrow S'$  ;
16                 fim
17             fim
18         fim
19      $T \leftarrow T * \beta$  ;
20 fim
    
```


Quatro movimentos tradicionais

- ▶ Move (M)
- ▶ Swap (S)
- ▶ Cadeia de Kempe Simples (KS)
- ▶ Cadeia de Kempe Estendida (KE)

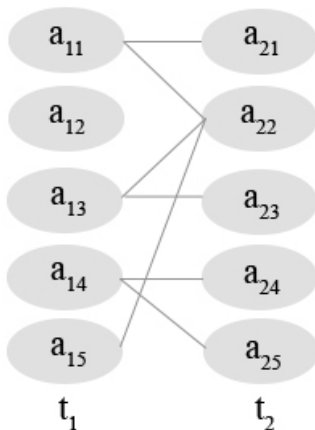
Sala X

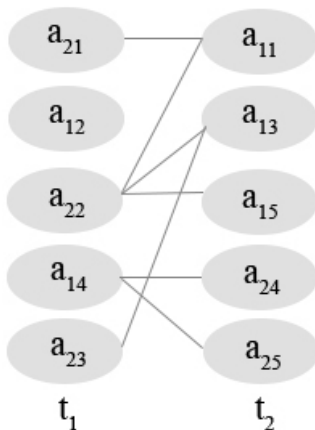
| | Segunda | Terça | Quarta | Quinta | Sexta |
|---------------|---------------|---------------|---------------|--------------|-----------------|
| 07:00 - 08:00 | | | | | |
| 08:00 - 09:00 | | | Mat. Discreta | | |
| 10:00 - 11:00 | | Programação 1 | | | |
| 11:00 - 12:00 | | | | Otim. Linear | |
| 12:00 - 13:00 | | | | | |
| 13:00 - 14:00 | Comp. Gráfica | Mat. Discreta | | | |
| 14:00 - 15:00 | | | | | |
| 15:00 - 16:00 | | | | | Teo. dos Grafos |
| 16:00 - 17:00 | | | | | Teo. dos Grafos |
| 17:00 - 18:00 | | | | | |

| Sala X | | | | | |
|---------------|---------|---------------|---------------|--------------|-----------------|
| | Segunda | Terça | Quarta | Quinta | Sexta |
| 07:00 - 08:00 | | | | | |
| 08:00 - 09:00 | | | | | |
| 10:00 - 11:00 | | Programação 1 | | | |
| 11:00 - 12:00 | | | | Otim. Linear | |
| 12:00 - 13:00 | | | | | |
| 13:00 - 14:00 | | | Mat. Discreta | | |
| 14:00 - 15:00 | | | | | |
| 15:00 - 16:00 | | | | | |
| 16:00 - 17:00 | | | | | Teo. dos Grafos |
| 17:00 - 18:00 | | | | | Teo. dos Grafos |



The diagram shows two blue curved arrows forming a cycle. One arrow points from the 'Otim. Linear' cell (Quinta, 11:00-12:00) to the 'Mat. Discreta' cell (Quarta, 13:00-14:00). The other arrow points from the 'Mat. Discreta' cell back to the 'Otim. Linear' cell, indicating a swap of these two courses.





- ▶ Cinco movimentos específicos propostos por Müller [2009]

Cinco movimentos específicos propostos por Müller [2009]

- ▶ Time Move (TM)
- ▶ Room Move (RM)
- ▶ Lecture Move (LM)
- ▶ Room Stability Move (RS)
- ▶ Minimum Working Days Move (MWD)

| Sala X | | | | | |
|---------------|---------|---------------|---------------|--------------|-----------------|
| | Segunda | Terça | Quarta | Quinta | Sexta |
| 07:00 - 08:00 | | | | | |
| 08:00 - 09:00 | | | | | |
| 10:00 - 11:00 | | Programação 1 | | | |
| 11:00 - 12:00 | | | | Otim. Linear | |
| 12:00 - 13:00 | | | | | |
| 13:00 - 14:00 | | | Mat. Discreta | | |
| 14:00 - 15:00 | | | | | |
| 15:00 - 16:00 | | | | | Teo. dos Grafos |
| 16:00 - 17:00 | | | | | Teo. dos Grafos |
| 17:00 - 18:00 | | | | | |

| Sala X | | | | | |
|---------------|---------|---------------|--------|--------------|-------|
| | Segunda | Terça | Quarta | Quinta | Sexta |
| 07:00 - 08:00 | | | | | |
| 08:00 - 09:00 | | | | | |
| 10:00 - 11:00 | | Programação 1 | | | |
| 11:00 - 12:00 | | | | Otim. Linear | |
| 12:00 - 13:00 | | | | | |
| 13:00 - 14:00 | | | | | |
| 14:00 - 15:00 | | | | | |
| 15:00 - 16:00 | | | | | |
| 16:00 - 17:00 | | | | | |
| 17:00 - 18:00 | | | | | |

| Sala Y | | | |
|---------------|---------|-------|--------|
| | Segunda | Terça | Quarta |
| 07:00 - 08:00 | | | |
| 08:00 - 09:00 | | | |
| 10:00 - 11:00 | | | |
| 11:00 - 12:00 | | | |
| 12:00 - 13:00 | | | |

A blue arrow points from the 'Programação 1' cell in Sala X to the 10:00 - 11:00 slot in Sala Y.

| Salas | Dia 1 | | Dia 2 | | Dia 3 | |
|----------------|-------|----|-------|----|-------|----|
| | P1 | P2 | P3 | P4 | P5 | P6 |
| r ₁ | | AT | TC | | | |
| r ₂ | | GT | AT | | | |
| r ₃ | AT | TC | | GT | | |
| r ₄ | GT | AV | | | | |
| r ₅ | | BP | | | | |

| Salas | Dia 1 | | Dia 2 | | Dia 3 | |
|----------------|-------|----|-------|----|-------|----|
| | P1 | P2 | P3 | P4 | P5 | P6 |
| r ₁ | | AT | GT | | | |
| r ₂ | | GT | AT | | | |
| r ₃ | AT | TC | | GT | | |
| r ₄ | TC | AV | | | | |
| r ₅ | | BP | | | | |

| Salas | Dia 1 | | Dia 2 | | Dia 3 | |
|----------------|-------|----|-------|----|-------|----|
| | p1 | p2 | p3 | p4 | p5 | p6 |
| r ₁ | | AT | TC | | | |
| r ₂ | | GT | AT | | | |
| r ₃ | AT | TC | | GT | | |
| r ₄ | GT | AV | | | | |
| r ₅ | | BP | | | | |

| Salas | Dia 1 | | Dia 2 | | Dia 3 | |
|----------------|-------|----|-------|----|-------|----|
| | p1 | p2 | p3 | p4 | p5 | p6 |
| r ₁ | | AT | TC | | | |
| r ₂ | | GT | AT | | | |
| r ₃ | AT | TC | GT | GT | | |
| r ₄ | | AV | | | | |
| r ₅ | | BP | | | | |

- ▶ 21 instâncias (comp01 ~ comp21) usadas no ITC-2007
- ▶ Os algoritmos foram implementados utilizando linguagem de programação C++, compilados com g++ versão 4.8.4 64 bits com a *flag* de otimização -O3 e testados em máquina com processador Intel I5-3570 3.4 GHz, 8 Gb de memória RAM, sistema operacional Ubuntu 14.04 LTS 64bits.
- ▶ Assim como feito no ITC-2007, o algoritmo foi executado 10 vezes para cada instância.
- ▶ Cada execução usou diferentes *seeds* para a geração de números aleatórios.
- ▶ Os valores de $f(S)$ representam apenas as violações das restrições fracas, dado que ao final de cada execução todas as soluções obtidas são viáveis.

| Parâmetro | Descrição | Valor |
|-----------|---|--------|
| maxIter | Número máximo de iterações do GRASP | 200 |
| α | Valor de <i>threshold</i> da LRC na construção da solução inicial | 0,15 |
| n | Número máximo de iterações sem melhora em <i>Hill Climbing</i> | 10000 |
| k | Número de vizinhos gerados por iteração no <i>Hill Climbing</i> | 10 |
| N_v | Número de vizinhos gerados em cada valor de temperatura no SA | 500 |
| t_i | Temperatura inicial no SA | 1,5 |
| t_f | Temperatura final no SA | 0,005 |
| β | Taxa de resfriamento no SA | 0,999 |
| seed | Seed de geração de números aleatórios | 1 ~ 10 |
| timeout | Tempo limite de execução. | 220 s |

| Instância | C_{GRASP} | | C_{Gulosa} | | $C_{ComAleatoriedade}$ | |
|-----------|-------------|---------|--------------|---------|------------------------|---------|
| | Construção | Após SD | Construção | Após SD | Construção | Após SD |
| comp01 | 623 | 50 | 330 | 25 | 324 | 40 |
| comp02 | 1075 | 267 | 521 | 190 | 676 | 236 |
| comp03 | 849 | 218 | 462 | 210 | 578 | 225 |
| comp04 | 1273 | 177 | 721 | 182 | 997 | 181 |
| comp05 | 1189 | 603 | 945 | 638 | 1030 | 696 |
| comp06 | 2342 | 254 | 1373 | 233 | 1779 | 283 |
| comp07 | 2615 | 248 | 1481 | 238 | 2059 | 238 |
| comp08 | 1522 | 184 | 883 | 150 | 1238 | 183 |
| comp09 | 1021 | 248 | 622 | 220 | 680 | 236 |
| comp10 | 1891 | 213 | 849 | 204 | 1264 | 215 |
| comp11 | 396 | 33 | 354 | 24 | 377 | 42 |
| comp12 | 1496 | 622 | 1752 | 673 | 1154 | 631 |
| comp13 | 1787 | 225 | 1088 | 215 | 1248 | 255 |
| comp14 | 1417 | 199 | 983 | 221 | 1139 | 204 |
| comp15 | 849 | 218 | 462 | 210 | 624 | 224 |
| comp16 | 1697 | 216 | 1053 | 233 | 1413 | 211 |
| comp17 | 1836 | 260 | 1023 | 253 | 1294 | 258 |
| comp18 | 600 | 220 | 558 | 204 | 549 | 212 |
| comp19 | 839 | 234 | 499 | 236 | 584 | 241 |
| comp20 | 3353 | 277 | 1809 | 286 | 2005 | 276 |
| comp21 | 1329 | 298 | 1033 | 258 | 966 | 291 |
| Média | 1428,5 | 250,7 | 895,3 | 243,0 | 1046,6 | 256,1 |

| Instância | GRASPK | GRASPK2 | GRASPHC | GRASPSD |
|-----------|--------|---------|---------|---------|
| comp01 | 5,5 | 5 | 17,9 | 25,8 |
| comp02 | 180,3 | 94,7 | 160,6 | 235,9 |
| comp03 | 157,3 | 102,4 | 196,9 | 201,6 |
| comp04 | 99,6 | 48,6 | 141,5 | 157,1 |
| comp05 | 537,8 | 383,2 | 589,4 | 488,3 |
| comp06 | 160,8 | 84,8 | 206,6 | 234,4 |
| comp07 | 142,3 | 51,7 | 211,9 | 227,5 |
| comp08 | 104,3 | 54,3 | 141 | 165,4 |
| comp09 | 167,5 | 119,8 | 206,2 | 227,7 |
| comp10 | 116,4 | 46,3 | 184,7 | 192,5 |
| comp11 | 0,3 | 0 | 12,9 | 14,8 |
| comp12 | 485 | 395,3 | 613,4 | 569,2 |
| comp13 | 139,8 | 87,4 | 174,4 | 207,6 |
| comp14 | 119,4 | 72 | 154,8 | 172,1 |
| comp15 | 157,3 | 102,4 | 196,9 | 201,6 |
| comp16 | 136 | 62,7 | 187,8 | 198,5 |
| comp17 | 164,6 | 103 | 227,4 | 234,7 |
| comp18 | 123 | 97,5 | 154,3 | 175,4 |
| comp19 | 148,6 | 86,2 | 187 | 211,9 |
| comp20 | 185,1 | 81,9 | 234,6 | 268,1 |
| comp21 | 211,3 | 135 | 255,6 | 270,3 |
| Média | 168,68 | 105,44 | 212,18 | 222,88 |

- ▶ Os testes foram divididos em duas categorias, considerando respectivamente estruturas de vizinhanças simples e combinadas, conforme descrito por Lü et al. [2011]
 - Estrutura de vizinhança simples é composta por um único movimento
 - Estrutura de vizinhança combinada é composta por dois ou três movimentos

- ▶ Vizinhança Combinada
 - União (\cup)
 - Sequência (\rightarrow)

União (\cup)

- ▶ Dada uma solução s e k diferentes movimentos $m_1 \sim m_k$
- ▶ Gerar todos os s vizinhos usando cada um dos k movimentos
- ▶ O melhor vizinho de s é escolhido e o processo é repetido até que não ocorra melhoria

Sequência (\rightarrow)

- ▶ Dada uma solução s e k diferentes movimentos $m_1 \sim m_k$
- ▶ SD é aplicado em s usando m_1
- ▶ O melhor vizinho é submetido a SD usando m_2 , e assim sucessivamente seguindo a sequência até m_k
- ▶ O processo é repetido até que não seja encontrada nenhuma melhoria.

Tabela 1

- ▶ Média sobre 50 execuções do SD
- ▶ Valores arredondados para baixo

| Instância | LM | M | S | RM | RS | KE | TM | KS | MWD |
|-----------|-----|------|-----|------|------|------|------|------|------|
| comp01 | 25 | 251 | 29 | 288 | 228 | 294 | 313 | 320 | 330 |
| comp02 | 190 | 209 | 304 | 408 | 499 | 476 | 485 | 467 | 507 |
| comp03 | 210 | 235 | 336 | 385 | 448 | 416 | 427 | 442 | 461 |
| comp04 | 182 | 195 | 349 | 396 | 581 | 626 | 669 | 676 | 666 |
| comp05 | 638 | 730 | 785 | 816 | 895 | 877 | 914 | 910 | 939 |
| comp06 | 233 | 287 | 388 | 573 | 884 | 1180 | 1243 | 1333 | 1362 |
| comp07 | 238 | 361 | 335 | 664 | 979 | 1238 | 1302 | 1426 | 1468 |
| comp08 | 149 | 171 | 338 | 455 | 668 | 692 | 783 | 843 | 826 |
| comp09 | 220 | 249 | 394 | 515 | 570 | 540 | 562 | 572 | 588 |
| comp10 | 204 | 243 | 369 | 611 | 714 | 644 | 726 | 825 | 815 |
| comp11 | 24 | 58 | 148 | 91 | 210 | 154 | 343 | 320 | 350 |
| comp12 | 673 | 1140 | 913 | 1109 | 1243 | 1742 | 1707 | 1694 | 1752 |
| comp13 | 215 | 226 | 407 | 436 | 772 | 1024 | 1013 | 1066 | 1088 |
| comp14 | 221 | 247 | 547 | 479 | 865 | 934 | 931 | 952 | 935 |
| comp15 | 210 | 235 | 336 | 385 | 448 | 416 | 427 | 442 | 461 |
| comp16 | 233 | 273 | 491 | 677 | 936 | 748 | 879 | 1007 | 1013 |
| comp17 | 253 | 312 | 377 | 639 | 708 | 810 | 925 | 947 | 993 |
| comp18 | 204 | 217 | 305 | 296 | 369 | 481 | 558 | 540 | 503 |
| comp19 | 236 | 262 | 340 | 380 | 423 | 463 | 485 | 472 | 487 |
| comp20 | 286 | 421 | 609 | 633 | 1316 | 1640 | 1716 | 1726 | 1805 |
| comp21 | 258 | 313 | 435 | 610 | 656 | 761 | 953 | 1001 | 1013 |

Gráfico 1

► Valores Normalizados

►
$$h = \frac{f - f_{\min}}{f_{\max} - f_{\min}}$$

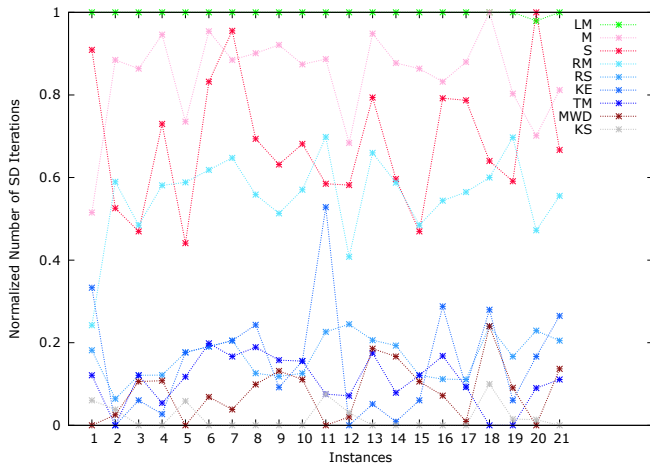


Gráfico 2

► Valores Normalizados

►
$$h = \frac{f - f_{\min}}{f_{\max} - f_{\min}}$$

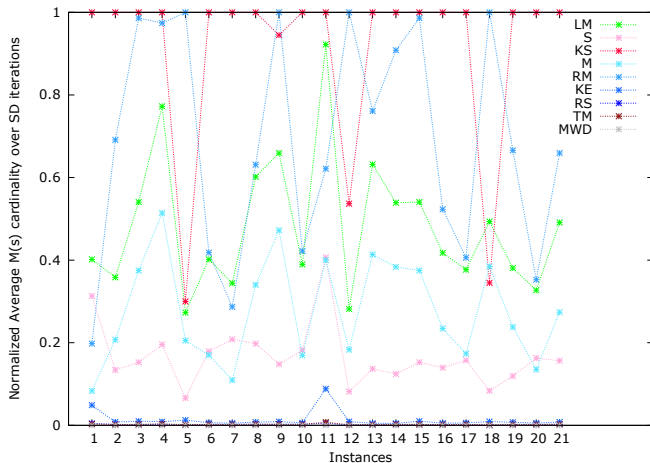
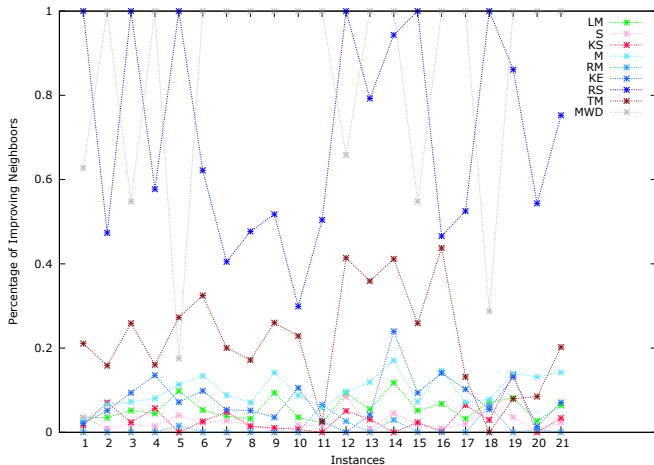


Gráfico 3

► Valores Normalizados

►
$$h = \frac{f - f_{\min}}{f_{\max} - f_{\min}}$$



- ▶ Comparando os resultados com Lü et al. [2011]
 - Com relação as estruturas de vizinhança simples similarmente implementadas
 - * Em Lü et al. [2011] os melhores resultados foram obtidos com *KempeSwap*, seguido por *SimpleMove*, *SimpleSwap* e *KempeMove*
 - * No nosso trabalho, **M** é o melhor, seguido por **S**, **KE** e **KS**
 - Nos dois trabalhos:
 - * **M** (*SimpleMove*) teve resultados melhores que **S** (*SimpleSwap*)
 - * **KE** (*KempeSwap*) teve resultados melhores que **KS** (*KempeMove*)

- ▶ Sete estruturas de vizinhança combinadas foram consideradas
 - $LM \cup RM$
 - $LM \cup TM$
 - $RM \cup TM$
 - $LM \cup RM \cup TM$
 - $LM \rightarrow KE$
 - $LM \rightarrow RS$
 - $LM \rightarrow MWD$
- ▶ **LM** foi usada em grande parte das estruturas de vizinhança combinada pois foi a estrutura de vizinhança simples que apresentou o melhor comportamento
- ▶ **LM**, **RM** e **TM** movimentos muito similares aos tradicionais **M** ou **S**, foram combinados usando (\cup)
- ▶ **LM** foi combinado em (\rightarrow) com outros movimentos diferentes do **M** e **S**

Tabela 2

- Média sobre 50 execuções do SD
- Valores arredondados para baixo

| Instance | LM → KE | LM → MWD | LM → RS | LM ∪ TM | LM ∪ RM | LM ∪ RM ∪ TM | RM ∪ TM |
|----------|---------|----------|---------|---------|---------|--------------|---------|
| comp01 | 24 | 25 | 25 | 25 | 25 | 25 | 310 |
| comp02 | 188 | 188 | 188 | 188 | 190 | 189 | 483 |
| comp03 | 211 | 211 | 211 | 209 | 210 | 210 | 417 |
| comp04 | 176 | 182 | 188 | 185 | 182 | 184 | 568 |
| comp05 | 636 | 638 | 641 | 638 | 638 | 638 | 912 |
| comp06 | 230 | 231 | 231 | 234 | 233 | 235 | 997 |
| comp07 | 232 | 236 | 231 | 238 | 240 | 241 | 971 |
| comp08 | 149 | 152 | 149 | 150 | 149 | 150 | 582 |
| comp09 | 224 | 221 | 221 | 222 | 222 | 221 | 559 |
| comp10 | 207 | 203 | 203 | 205 | 207 | 206 | 629 |
| comp11 | 26 | 24 | 21 | 24 | 24 | 24 | 327 |
| comp12 | 629 | 673 | 679 | 677 | 680 | 678 | 1707 |
| comp13 | 216 | 212 | 216 | 215 | 215 | 215 | 891 |
| comp14 | 221 | 213 | 227 | 221 | 224 | 226 | 902 |
| comp15 | 211 | 211 | 211 | 209 | 210 | 210 | 417 |
| comp16 | 222 | 229 | 230 | 230 | 234 | 233 | 721 |
| comp17 | 250 | 258 | 254 | 258 | 255 | 254 | 873 |
| comp18 | 199 | 201 | 204 | 204 | 204 | 204 | 558 |
| comp19 | 235 | 236 | 235 | 236 | 234 | 235 | 468 |
| comp20 | 280 | 280 | 274 | 287 | 286 | 286 | 1337 |
| comp21 | 259 | 264 | 258 | 262 | 257 | 261 | 753 |

Gráfico 4

► Valores Normalizados

►
$$h = \frac{f - f_{\min}}{f_{\max} - f_{\min}}$$

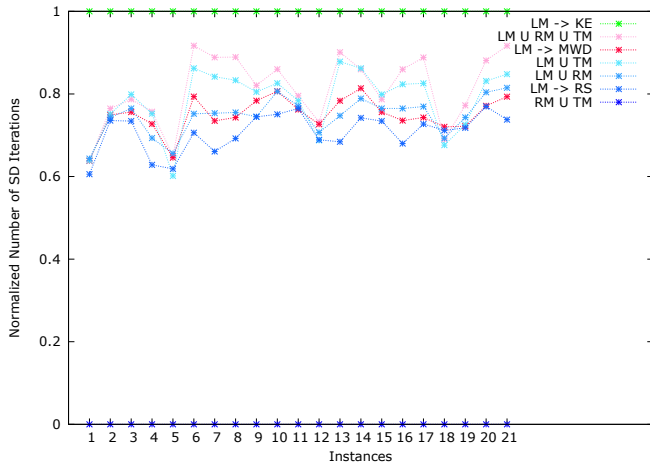


Gráfico 5

► Valores Normalizados

►
$$h = \frac{f - f_{\min}}{f_{\max} - f_{\min}}$$

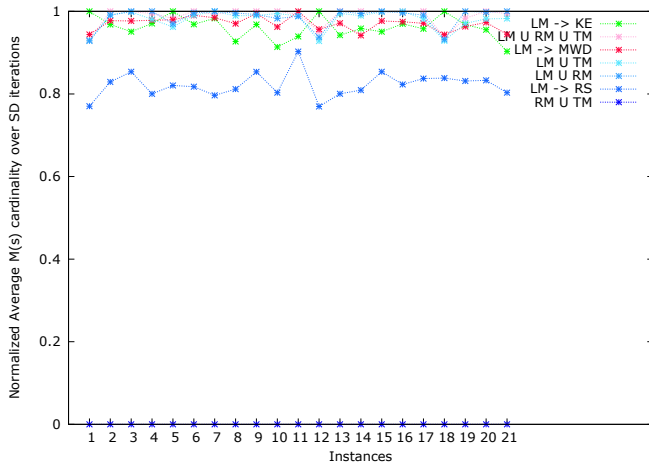
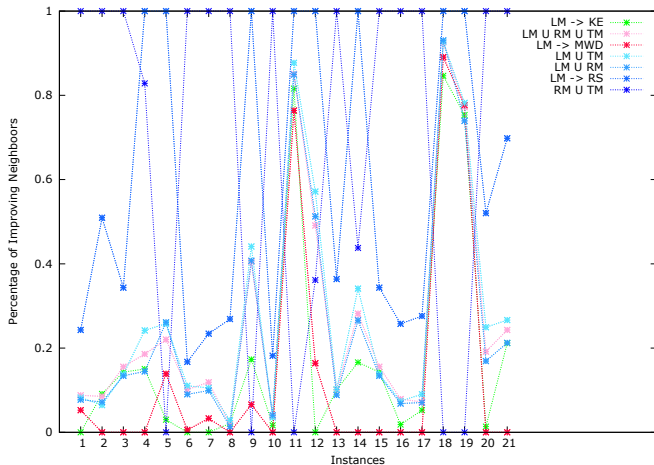


Gráfico 6

► Valores Normalizados

►
$$h = \frac{f - f_{\min}}{f_{\max} - f_{\min}}$$



| | Müller | Lü & Hao | Kiefer | GRASPK | GRASPK2 | GMF-D |
|--------|--------|----------|--------|--------|---------|-------|
| comp01 | 5 | 5 | 5 | 5,5 | 5 | 5 |
| comp02 | 61,3 | 60,6 | 41,5 | 180,3 | 55,1 | 61,5 |
| comp03 | 94,8 | 86,6 | 71,7 | 157,3 | 84,95 | 85,2 |
| comp04 | 42,8 | 47,9 | 35,1 | 99,6 | 40,8 | 39,5 |
| comp05 | 343,5 | 328,5 | 305,2 | 537,8 | 328,45 | 342,5 |
| comp06 | 56,8 | 69,9 | 47,8 | 160,8 | 58,35 | 58,8 |
| comp07 | 33,9 | 28,2 | 14,5 | 142,3 | 28,7 | 28,8 |
| comp08 | 46,5 | 51,4 | 41,0 | 104,3 | 45,45 | 44,6 |
| comp09 | 113,1 | 113,2 | 102,8 | 167,5 | 107,9 | 106,6 |
| comp10 | 21,3 | 38 | 14,3 | 116,4 | 25,05 | 25,1 |
| comp11 | 0 | 0 | 0 | 0,3 | 0 | 0 |
| comp12 | 351,6 | 365 | 319,4 | 485 | 353,3 | 350,9 |
| comp13 | 73,9 | 76,2 | 60,7 | 139,8 | 76,75 | 75,6 |
| comp14 | 61,8 | 62,9 | 54,1 | 119,4 | 61 | 60,4 |
| comp15 | 94,8 | 87,8 | 72,1 | 157,3 | 84,95 | 86,2 |
| comp16 | 41,2 | 53,7 | 33,8 | 136 | 44,4 | 44,1 |
| comp17 | 86,6 | 100,5 | 75,7 | 164,6 | 85 | 85,3 |
| comp18 | 91,7 | 82,6 | 66,9 | 123 | 84,85 | 86,0 |
| comp19 | 68,8 | 75 | 62,6 | 148,6 | 69,7 | 72,3 |
| comp20 | 34,3 | 58,2 | 27,2 | 185,1 | 45,95 | 43,5 |
| comp21 | 108 | 125,3 | 97,0 | 211,3 | 108,95 | 111,3 |
| Média | 87,22 | 91,26 | 73,73 | 168,68 | 85,46 | 86,34 |

- ▶ O grupo de *Timetabling* da UFES abordou o PTHU utilizando a terceira formulação do ITC-2007 em diferentes abordagens de pesquisa
- ▶ Para uma comparação fiel ao ambiente computacional onde os algoritmos submetidos no campeonato foram testados, utilizou-se as mesmas instâncias e tempo equivalente a execução em uma máquina utilizada no ITC-2007
- ▶ Com os resultados obtidos, pode-se observar que a utilização da meta-heurística GRASP combinada com uma etapa de construção e uma busca local (*Simulated Annealing*) comprovadamente eficientes, proporcionam melhorias significativas nos resultados.

- ▶ Além disso, destaca-se que a busca local usa movimentos que proporcionam resultados de destaque e o código-fonte foi reestruturado para garantir mais eficiência na execução
- ▶ Caso o algoritmo tivesse participado do ITC-2007, ocuparia a primeira colocação na classificação final da competição.
- ▶ Como trabalhos futuros pretende-se implementar métodos híbridos (Meta-heurísticas e Exatos) para obtenção de limitantes inferiores (*Lower Bounds*).
 - Geração de Coluna
 - Relaxação Lagrangeana

- ▶ Kampke, E. H.; Scheideger, L. M.; Mauri, G. R.; Boeres, M. C. S. (2019). A network flow based construction for a grasp+sa algorithm to solve the university timetabling problem. Lecture Notes in Computer Science, p. 215–231.
- ▶ Kampke, E. H.; Segatto, E. A.; Boeres, M. C. S., Rangel; M. C.; Mauri, G. R. (2017). Neighborhood analysis on the university timetabling problem. Lecture Notes in Computer Science, p.148–164.
- ▶ Lü, Z.; Hao, J.-K. (2010). Adaptive tabu search for course timetabling. European Journal of Operational Research, 200(1):235–244.
- ▶ Lü, Z.; Hao, J.-K.; Glover, F. (2011). Neighborhood analysis: A case study on curriculum-based course timetabling. Journal of Heuristics, 17(2):97–118.

- ▶ Müller, T. (2009). Itc2007 solver description: a hybrid approach. *Annals of Operations Research*, 172(1):429.
- ▶ Rocha, W. S. (2013). Algoritmo grasp para o problema de tabela-horário de universidades. Master's thesis, Universidade Federal do Espírito Santo.
- ▶ Schaerf, A. (1995). A survey of automated timetabling. *Artificial Intelligence Review*, 13:87–127.
- ▶ Segatto, E. A.; Boeres, M. C. S.; Rangel, M. C.; Kampke, E. H. (2015). Um algoritmo grasp com cadeia de kempe aplicado ao problema de tabela-horário para universidades. *Anais do XLVII SBPO-Simpósio Brasileiro de Pesquisa Operacional*, p.2643–2654.

Obrigado!

Edmar Hell Kampke

Maria Claudia Silva Boeres

edmar.kampke@ufes.br

boeres@inf.ufes.br

*Laboratório de Otimização – LabOtim
Universidade Federal do Espírito Santo – UFES*

12 de Setembro de 2023



aaaa.

aaaa.

Kampke, E. H., Scheideger, L. M., Mauri, G. R., e Boeres, M. C. S.

(2019). A network flow based construction for a grasp+sa algorithm to solve the university timetabling problem. In *Lecture Notes in Computer Science*, p. 215–231.

Kampke, E. H., Segatto, E. A., Boeres, M. C. S., Rangel, M. C., e Mauri, G. R. (2017). Neighborhood analysis on the university timetabling problem. In *Lecture Notes in Computer Science*, p. 148–164.

Lü, Z. e Hao, J.-K. (2010). Adaptive tabu search for course timetabling. *European Journal of Operational Research*, 200(1): 235–244.

Lü, Z., Hao, J.-K., e Glover, F. (2011). Neighborhood analysis: A case study on curriculum-based course timetabling. *Journal of Heuristics*, 17(2):97–118.

Müller, T. (2009). Itc2007 solver description: a hybrid approach.

Annals of Operations Research, 172(1):429.

Rocha, W. S. (2013). Algoritmo grasp para o problema de tabela-horário de universidades. Master's thesis, Universidade Federal do Espírito Santo.



~~Schaerf, A. (1995). A survey of automated timetabling. *Artificial Intelligence Review*, 13:87–127.~~

Segatto, E. A., Boeres, M. C. S., Rangel, M. C., e Kampke, E. H. (2015). Um algoritmo grasp com cadeia de kempe aplicado ao problema de tabela-horário para universidades. In *Anais do XLVII SBPO-Simpósio Brasileiro de Pesquisa Operacional*, number 1, p. 2643–2654.