

## 2

### Descrição do Problema

#### 2.1

#### International Timetabling Competition (ITC)

##### 2.1.1

##### Primeira edição: ITC 2002

No ano de 2002, como um evento paralelo ao PATAT 2002, foi promovida a primeira edição de uma competição internacional de programação de horários, a *International Timetabling Competition* (Paechter et al., 2009), organizada pelo projeto *Metaheuristics Network* (Dorigo, 2009).

A competição procurava incentivar a pesquisa no campo de programação de horários e estabelecer um problema *benchmark* que fosse amplamente aceito pela comunidade de programação de horários. O problema escolhido foi o Problema de Programação de Horários de Cursos (PPHC), que discutimos na seção 2.2. O modelo elaborado para a competição foi baseado em uma simplificação do problema prático de programação de horários encontrado na Napier University, em Edimburgo, Escócia, e as instâncias utilizadas foram produzidas por um gerador de instâncias desenvolvido por Paechter (Paechter, 2009).

##### 2.1.2

##### Segunda edição: ITC 2007

Devido ao sucesso alcançado pela primeira ITC, no ano de 2007 foi promovida a segunda edição da competição, *Second International Timetabling Competition* (ITC 2007) (McCollum, 2009a), como um evento paralelo ao PATAT 2008. Desta vez, a organização ficou por conta do projeto EventMAP (McCollum, 2009b).

Diferente da primeira edição, em que apenas um problema foi proposto, na segunda edição da ITC foram estabelecidas três categorias, cada uma dedicada a um problema: Programação de Exames, Programação de Horários de Cursos Pós-Matrícula e Programação de Horários de Cursos baseada em Currículos.

O Problema de Programação de Horários de Cursos Pós-Matrícula (PPHCPM), que definimos na seção 2.3, é uma extensão do PPHC da primeira edição da competição. A segunda edição também trouxe uma mudança considerável na forma de avaliar as soluções fornecidas pelos algoritmos, o que discutimos na seção 2.3.3.

## 2.2

### Problema de Programação de Horários de Cursos (PPHC) da ITC 2002

Na primeira edição da ITC, os competidores foram desafiados a projetar um algoritmo para o problema de Programação de Horários de Cursos (PPHC). A definição fornecida pela organização da competição tratava do problema de alocar os eventos dos cursos em períodos e salas, sem burlar as restrições estabelecidas.

Um aspecto importante da definição do problema é que os estudantes escolhem a priori quais eventos desejam assistir, e estas escolhas devem ser respeitadas na construção da tabela de horários, que é elaborada somente após os estudantes fazerem suas escolhas.

#### 2.2.1

##### Dados das Instâncias do PPHC

Para qualquer das instâncias consideradas, o conjunto de períodos disponíveis para a alocação dos eventos, que denotamos por  $\mathcal{P}$ , é o mesmo. Trata-se de um conjunto de 45 períodos, representando a união dos conjuntos de 9 períodos (de uma hora cada) de cada um dos 5 dias da semana.

Quando for conveniente, poderemos nos referir também ao conjunto de dias, que denotamos por  $\mathcal{D}$ . Para representar o dia em que ocorre um determinado período  $p \in \mathcal{P}$ , utilizamos a notação  $\mathcal{D}(p)$ , e para representar o conjunto de períodos que compõem um dia  $d \in \mathcal{D}$ , utilizamos a notação  $\mathcal{P}^d$ .

Cada problema consiste das informações a seguir:

- $\mathcal{E}$ , um conjunto de eventos a serem alocados aos períodos e salas;
- $\mathcal{R}$ , um conjunto de salas nas quais os eventos ocorrem, cada uma com uma dada capacidade de ocupantes;
- $\mathcal{F}$ , um conjunto de recursos que são *disponibilizados* pelas salas e são *requeridos* pelos eventos;
- $\mathcal{S}$ , um conjunto de estudantes que *assistem* a várias combinações diferentes de eventos.

Quando for conveniente, utilizaremos a notação a seguir para representar os relacionamentos básicos entre estes conjuntos:

- $Capacity(r)$ , a capacidade de uma sala  $r \in \mathcal{R}$ ;
- $\mathcal{F}^r$ , o conjunto de recursos disponibilizados por uma sala  $r \in \mathcal{R}$ ;
- $\mathcal{F}^e$ , o conjunto de recursos requeridos por um evento  $e \in \mathcal{E}$ ;
- $\mathcal{E}^s$ , o conjunto de eventos assistidos por um estudante  $s \in \mathcal{S}$ ;
- $\mathcal{S}^e$ , o conjunto de estudantes que assistem um evento  $e \in \mathcal{E}$ .

Note que para evitar ambiguidade, ao mencionar um conjunto  $\mathcal{F}^i$  por exemplo, sempre explicitaremos a origem de  $i$ . Se  $i \in \mathcal{R}$ , por exemplo, fica fácil inferir que trata-se do conjunto de recursos disponibilizados pela sala  $r$ .

Informações detalhadas sobre o formato de arquivo utilizado para representar as instâncias, entre outras informações, podem ser obtidas no *site* da ITC 2002 (Paechter et al., 2009).

### Pré-processamento de Instâncias

Alguns pré-processamentos simples podem ser feitos sobre as instâncias para inferir informações úteis para a resolução do problema. Em particular, dois tipos de conjuntos podem ser inferidos a partir dos dados básicos providos pela instância:

- $\mathcal{R}^e$ , o conjunto salas *apropriadas* para um evento  $e \in \mathcal{E}$ ;
- $\Delta(e)$ , o conjunto dos eventos que não podem ser alocados ao mesmo período que um evento  $e \in \mathcal{E}$ , também chamado de conjunto *adjacência* de  $e$ .

Para descobrir se uma sala é apropriada, ou seja, se pode ser utilizada por um dado evento, basta verificar duas coisas: se a capacidade da sala é suficiente para a quantidade de alunos que assistem ao evento em questão e se todos os recursos que são requeridos por este evento estão disponíveis nesta sala. Construimos assim então os conjuntos  $\mathcal{R}^e$  para cada evento  $e \in \mathcal{E}$ :

$$\mathcal{R}^e = \{r \in \mathcal{R} : Capacity(r) \geq |\mathcal{S}^e|, \mathcal{F}^e \subseteq \mathcal{F}^r\}, \quad e \in \mathcal{E}$$

Alguns eventos não podem ser alocados no mesmo período que um dado evento, fazendo parte do conjunto *adjacência* deste evento. Este impedimento é criado quando os eventos têm estudantes em comum, o que faz com que sua ocorrência simultânea impeça que um estudante os assista. Construimos assim os conjuntos adjacência  $\Delta(u)$  para cada evento  $u \in \mathcal{E}$ :

$$\Delta(u) = \{v \in \mathcal{E} : v \neq u, \mathcal{S}^v \cap \mathcal{S}^u \neq \emptyset\}, \quad u \in \mathcal{E}$$

### 2.2.2

#### Restrições Fortes do PPHC

O objetivo ao resolver o PPHC é inserir cada evento na tabela de horários, ou seja, atribuir cada um dos eventos  $e \in \mathcal{E}$  a um período  $p \in \mathcal{P}$  e uma sala  $r \in \mathcal{R}$ , observando um conjunto de *restrições fortes*:

- **NoStudentClash**: Nenhum estudante deve ser obrigado a assistir mais de um evento ao mesmo tempo;
- **RoomFitting**: Para cada alocação de um evento a uma sala, a capacidade da sala deve ser suficiente para comportar todos os estudantes que assistem ao evento, assim como disponibilizar todos os recursos que são requeridos pelo evento;
- **NoRoomClash**: No máximo um evento pode ser alocado a cada sala, em cada período.

Uma tabela de horários em que haja quaisquer violações destas restrições é dita *inviável*.

### 2.2.3

#### Restrições Fracas do PPHC

Há também um conjunto de *restrições fracas* que devem evitadas para que a tabela de horários seja confortável para os seus usuários, os estudantes. No entanto, estas restrições fracas podem ser violadas sem que a tabela de horários torne-se inviável. São elas:

- **NoEndOfDayEvent**: Nenhum estudante deveria ter que assistir um evento no último período do dia;
- **NoEventSequence**: Nenhum estudante deveria ter que assistir mais que dois eventos em sequência;
- **NoSingleEvent**: Nenhum estudante deveria ter que assistir um único evento no dia.

A escolha das restrições fracas foi feita de forma a ser representativa de três classes distintas de restrições (Rossi-Doria et al., 2003). Uma violação da restrição *NoEndOfDayEvent* pode ser verificada apenas examinando uma alocação individual, sem conhecimento do restante da tabela de horários. Já uma violação da restrição *NoEventSequence* pode ser verificada durante a construção da solução, levando em consideração os períodos vizinhos. Por último, uma violação da restrição *NoSingleEvent* somente pode ser confirmada após a construção da tabela de horário ter sido completada, com todos os eventos atribuídos a períodos.

#### 2.2.4

##### Avaliação de Soluções do PPHC

Em relação ao julgamento da qualidade das soluções, foi decidido pela organização que o critério de avaliação seria unicamente a soma das penalidades associadas às violações de restrições fracas. Um algoritmo que não fosse capaz de gerar soluções sem violações de restrições fortes dentro do tempo estabelecido não poderia participar da competição, uma vez que as soluções fornecidas seriam prontamente desqualificadas. Uma razão para isso era evitar o problema de decidir como comparar duas soluções com diferentes números de violações de restrições fortes e fracas.

A medida de qualidade de uma solução do PPHC é composta por um único indicador:

- **SoftCost**: soma de todas as penalidades associadas a violações de restrições fracas.

Considera-se que cada violação de restrição fraca, independente de sua natureza, é igualmente prejudicial à qualidade da tabela de horários. Portanto, as penalidades são unitárias e o valor de uma solução é simplesmente a quantidade de violações de restrições fracas.

Para não restringir demasiadamente a participação dos desafiantes (uma vez que as soluções inviáveis são descartadas), as instâncias da competição foram elaboradas de modo a facilitar a resolução das restrições fortes, o que adicionou um viés à competição: o que observou-se ao seu final foi que a maioria das idéias produzidas durante o evento estavam relacionada com a resolução das restrições fracas. Os algoritmos mais bem-sucedidos utilizaram-se de uma abordagem de dois estágios, onde no estágio inicial utilizam-se várias heurísticas construtivas resolver rapidamente as restrições fortes, e o segundo estágio (que toma a maior parte do tempo) é dedicado a minimizar as violações de restrições fracas.

Entretanto, conforme apontado em (Lewis & Paechter, 2007), sabemos que na prática da programação de horários em instituições de ensino nem sempre é tão fácil encontrar soluções viáveis para o problema. De fato o problema de encontrar uma tabela de horários viável é NP-difícil e sua dificuldade não deve ser subestimada.

### 2.3

#### Problema de Programação de Horários de Cursos Pós-Matrícula (PPHCPM) da ITC 2007

Como mencionamos anteriormente, o PPHCPM da ITC 2007 é uma extensão do PPHC da ITC 2002. O modelo foi acrescido de duas novas

restrições fortes, visando aproximá-lo ainda mais dos problemas encontrados na prática da programação de horários em instituições de ensino.

Um relatório completo sobre o modelo do PPHCPM utilizado na ITC 2007 foi elaborado pela organização da competição (Lewis et al., 2007). Além de definir o modelo, o relatório descreve os motivos das alterações em relação ao modelo anterior, as mudanças na avaliação de soluções e as limitações do modelo proposto em relação a sua aplicação prática.

### 2.3.1

#### Dados das Instâncias do PPHCPM

Os dados das instâncias são os mesmos do PPHC (ver seção 2.2.1), acrescidos dos dados a seguir:

- $\mathcal{P}^e$ , conjuntos de períodos *apropriados* para um evento  $e \in \mathcal{E}$  (nem todo período é apropriado para todo evento);
- $\mathbf{Prec}(e)$ , um conjunto de eventos que devem *preceder* um evento  $e \in \mathcal{E}$  (quando  $e$  não tem nenhuma restrição de precedência,  $\mathbf{Prec}(e) = \emptyset$ ).

Por conveniência, utilizamos também a seguinte notação:

- $\mathcal{E}^p$ , conjuntos de eventos para os quais um período  $p \in \mathcal{P}$  é apropriado.

Informações detalhadas sobre o formato de arquivo utilizado para as instâncias podem ser obtidas no *site* da ITC 2007 (McCollum, 2009a).

#### Pré-processamento de Instâncias

O conjunto de salas apropriadas,  $\mathcal{R}^e$ , pode ser construído da mesma forma que no PPHC (ver seção 2.2.1). Já em relação ao conjunto  $\Delta(e)$ , consideramos aqui uma condição adicional. Quando dois eventos têm uma relação de precedência, eles também não devem ser alocados para um mesmo período, logo alteramos a construção de  $\Delta(e)$ , incluindo esta consideração:

$$\Delta(u) = \{v \in \mathcal{E} : v \neq u, (\mathcal{S}^v \cap \mathcal{S}^u \neq \emptyset \vee v \in \mathbf{Prec}(u) \vee u \in \mathbf{Prec}(v))\}, \quad u \in \mathcal{E}$$

### 2.3.2

#### Restrições do PPHCPM

Duas restrições fortes foram adicionadas ao modelo, além das restrições do PPHC (ver seção 2.2.2):

- **PeriodFitting**: Cada evento deve ser alocado a um período definido previamente como apropriado para este evento;

- **Precedence**: Eventos com restrições de precedência devem ocorrer na ordem correta na semana.

Em relação às restrições fracas, não houve alterações, continuando a valer as mesmas restrições listadas para o PHC (ver seção 2.2.3).

### 2.3.3

#### Avaliação de Soluções do PPHCPM

Com a adição das novas restrições fortes, a organização considerou que não seria razoável utilizar a mesma forma de avaliação utilizada na ITC 2002, uma vez que tornou-se mais difícil encontrar soluções sem quaisquer violações de restrições fortes. Portanto, foi definido um novo método de avaliação, capaz de comparar duas soluções com diferentes números de violações de restrições fortes e fracas.

Em relação à *viabilidade*, uma solução que apresenta violações de restrições fortes continua a ser considerada *inviável*. O que muda é a adoção do conceito de *validade* de soluções. Uma solução que não apresenta violações viáveis é considerada *válida*, mesmo que nesta solução alguns eventos estejam desalocados, ou seja, sem atribuição a um período e sala. Uma solução viável, portanto, é uma solução válida onde não há eventos desalocados.

A medida de qualidade de uma solução do PPHCPM é composta por dois indicadores. Discutimos anteriormente (ver seção 2.2.4) o *SoftCost*, e agora apresentamos o indicador introduzido pela ITC 2007:

- **DtF** (*Distance to Feasibility*): este valor é uma medida da *quantidade de inviabilidade* de uma solução. É calculado somando-se as quantidades de alunos que assistem a cada um dos eventos desalocados.

Para comparar duas soluções do PPHCPM, primeiramente compara-se a *DtF*. Se os valores forem diferentes, a solução de menor *DtF* é tomada como a melhor (independente do *SoftCost*). Se os valores de *DtF* forem iguais, então compara-se o *SoftCost*, sendo considerada melhor a solução de custo mais baixo.