CORDENONSI, A. Z. ; ARAMBURU, L. G. C. ; ALMANCA, L. *Resolução do Problema de Quadro de Horários Através de um Algoritmo de Satisfação de Restrições*. In: VIII Simpósio de Informática e III Mostra de Software Acadêmico, 2003, Uruguaina - RS, 2003.

Resolução do Problema de Quadro de Horários Através de um Algoritmo de Satisfação de Restrições

Leonardo G.C. Aramburu[1]

Leonardo Almaça[2]

Andre Zanki Cordenonsi[3]

Resumo *– Este trabalho apresenta um algoritmo de satisfação de restrições para a resolução do problema de quadro de horários, considerando uma instituição com aulas de quatro períodos e que não haja problemas de deslocamento para os professores (aulas fisicamente no mesmo espaço). O algoritmo proposto apresenta uma solução inicial que pode ser utilizada como ponto de partida para outros tipos de algoritmos de melhoramento, tais como algoritmos genéticos ou meméticos.*

Palavras Chave – *Problema do Quadro de Horários, Satisfação de Restrições.*

I. INTRODUÇÃO

Este trabalho abrange uma proposta de solução para o problema de horários escolares. Este problema faz parte da classe de problemas denominada *Timetabling Problem* (TTP). As dificuldades do trabalho estão na alocação de professores nos horários oferecidos por uma instituição, levando-se em conta as devidas restrições, ou seja, limitações quanto à disponibilidade dos professores ou das salas de aula. Um sistema automático para resolução desta classe de problemas é utilizado para oferecer maior praticidade, diminuição de esforço físico e do tempo dispensado, caso este trabalho fosse realizado manualmente, tendo ainda a vantagem de oferecer diversos resultados para análise posterior da pessoa responsável pela elaboração dos horários [1].

O *Timetabling Problem* é a designação genérica para uma classe de problemas combinatórios que consistem em construir quadros de horários para uma série de atividades atendendo a um determinado conjunto de restrições. Cada atividade pode ou não envolver a utilização de recursos. Um exemplo comum é o problema de determinar os horários de aulas para as turmas de uma instituição de ensino. Neste caso as restrições seriam as limitações quanto à disponibilidade dos professores ou das salas de aula.

II. PROBLEMA DO QUADRO DE HORÁRIOS

O tipo de restrições envolvidas no problema varia bastante, dando origem a uma série de variantes do TTP. Essas restrições podem envolver a disponibilidade dos recursos, preferências pessoais, características desejadas no *layout* do quadro de horários, distância entre os diversos locais de aula, a existência de mais de um professor designado para uma mesma disciplina; e diversas outras restrições que dependem da instituição em questão.

Se considerarmos a complexidade das restrições e o número de combinações possíveis (explosão combinatória), a utilização de algoritmos ótimos normalmente torna-se inviável quando o tamanho da entrada para o problema é muito grande ou

[1]

[2]

[3] Centro Universitário Franciscano - UNIFRA, Rua dos Andradas 1614 – 97010.032, (55) 220 1235, andrezc@unifra.br

quando as restrições são demasiadamente complicadas. De fato, pode-se observar na bibliografia que grande parte das abordagens é feita através de algoritmos heurísticos.

Outra característica comum aos algoritmos normalmente utilizados é a sua especificidade, o que dificulta a utilização do mesmo algoritmo para se resolver variantes um pouco diferentes do problema.

A. Principais variações de TTP

- *School Timetabling*: agendamento semanal em escolas de 2º grau;
- *Course Timetabling*: agendamento semanal para cursos universitários;
- *Examination Timetabling*: agendamento dos exames de um conjunto de cursos universitários.

B. Formalização matemática

Para formalizar o sistema, se deve levar em conta os aspectos relevantes ao problema em questão. O estudo de caso realizado neste trabalho tem como abordagem a construção de uma grade de horários para o um curso universitário.

Inicialmente, é possível formalizar o problema matematicamente, usando combinações a priori admissíveis, referente aos docentes, disciplinas, semestres e horários de segunda-feira à sexta-feira, considerando as seguintes restrições:

- distância entre sala de aulas: não será considerada, pois todos os aposentos encontram-se fisicamente no mesmo prédio;
- capacidade: para efeito de cálculo, todas as salas de aula serão consideradas de mesmo tamanho e sua capacidade será suficiente para conter qualquer turma de qualquer semestre do curso analisado.

Consideremos, então, P_{ij} representando a designação do professor *i* para disciplina *j*. Onde:

$$j = 1...n \rightarrow \textit{número disciplinas}$$
$$i = 1...m \rightarrow \textit{número professores} \qquad (1)$$
$$P_{ij} = \begin{cases} 1, \textit{se o prof. i for designado para a disciplina j} \\ 0, \textit{caso contrário} \end{cases}$$

Consideremos S_{kj} a representação da designação da disciplina *j* para o semestre *k*. Onde:

$$j = 1...n \rightarrow \textit{número disciplinas}$$
$$k = 1...o \rightarrow \textit{número semestres} \qquad (2)$$
$$S_{kj} = \begin{cases} 1, \textit{se a disciplina j for designada para o semestre k} \\ 0, \textit{caso contrário} \end{cases}$$

Consideremos PS_{ik} como uma representação auxiliar que é construída através das combinações P_{ij} e S_{kj}, que mostra designação do professor *i* para o semestre *k*.

$$i = 1...m \rightarrow \textit{número professores}$$
$$k = 1...o \rightarrow \textit{número semestres} \qquad (3)$$
$$PS_{ik} = \begin{cases} 1, \textit{se o professor i for designado para o semestre k} \\ 0, \textit{caso contrário} \end{cases}$$

Consideremos T_j a representação da carga horária da disciplina *j*. Onde:

$$j = 1...n \rightarrow \textit{número disciplinas} \qquad (4)$$
$$T_j = \begin{cases} 2, \textit{quando a disciplina j tiver 2 horas/aula} \\ 3, \textit{quando a disciplina j tiver 3 horas/aula} \\ r, \textit{quando a disciplina j tiver r horas/aula} \end{cases}$$

Consideremos MH_{jhk} a representação onde a disciplina *j* for designada para o horário *h* do semestre *k*.

$$j = 1...n \rightarrow \textit{número disciplinas}$$
$$h = 1...p \rightarrow \textit{número horários} \qquad (5)$$
$$k = 1...o \rightarrow \textit{número semestres}$$
$$MH_{jhk} = \begin{cases} 1, \textit{se a disciplina j for designada para o} \\ \textit{o horário h do semestre k} \\ 0, \textit{caso contrário} \end{cases}$$

C. Restrição do Problema

Um professor *i* não pode ministrar duas disciplinas *j* e *j'* no mesmo horário *h*. Consideremos *i'* um professor qualquer, *j'* uma disciplina qualquer e $P_{i'j'} = 1$.

$$Se \quad MHj'h'k' = 1 \quad então$$
$$MH_{jhk} = 0 \; para\ todo\ h = h'\ e\ k = k'$$

III. SATISFAÇÃO DE RESTRIÇÕES

Muitos problemas de Inteligência Artificial (IA) podem ser vistos como problemas de satisfação de restrições em que o objetivo é descobrir algum estado de problema que satisfaça a um determinado conjunto de restrições. Exemplos deste tipo de problema incluem a criptoaritmética e em muitos problemas de interpretação da percepção do mundo real. As tarefas de projeto também podem ser vistas como problemas de satisfação de restrições em que um projeto precisa ser criado dentro de limites fixos de tempo, custo e material.

O termo restrições (*constraints*) e propagação de restrições (*constraint propagation*) têm sido amplamente utilizados em Inteligência Artificial para designar diferentes tipos de técnicas. Normalmente, as implementações que utilizam restrições tratam problemas combinatoriais com o objetivo de restringir o espaço de busca, minimizando o gasto de memória e de tempo de processamento, conseguindo com isso maior eficiência. O uso de restrições foi proposto para diminuir o espaço de busca na execução dos programas. A execução utiliza um mecanismo de controle que elimina os caminhos inválidos, isto é, caminhos que provavelmente não levem ao resultado final. Deste modo, minimiza-se o processamento desnecessário e o retrocesso. Várias linguagens em lógica com restrições foram criadas. Elas têm como principal aplicação problemas combinatoriais, como o escalonamento de horários e alocação de recursos.

Devido a estas características, a lógica com restrições pode ser aplicada eficientemente e diretamente em problemas de horários escolares. Um algoritmo de Satisfação de Restrições para o problema de Horários Escolares necessita de: (1) uma função heurística, que avalia um determinado horário frente ao conjunto de todos os horários criados; e (2) um conjunto de restrições pré-definido que, neste caso específico, pode ser descrito como todas as restrições definidas por cada professor pertencente ao quadro de horários.

IV. ALGORITMO DE RESTRIÇÕES PROPOSTO

Inicialmente, se pode formalizar o problema, usando as combinações admissíveis, referentes aos docentes, disciplinas, semestres e horários de segunda-feira à sexta-feira, com quatro períodos em cada dia totalizando vinte horas semanais, sem levar em consideração a distância de uma sala para outra e nem a sua capacidade de alunos, considerando que todas estão fisicamente no mesmo prédio e possuem o mesmo tamanho.

Este estudo considera que cada aula tem seu tempo pré-definido, bem como seu intervalo. Todas as turmas podem utilizar quaisquer salas de aula existentes no prédio, pois as mesmas serão consideradas com tamanho único e características iguais. Desse modo, não será tratado neste trabalho o problema relativo a espaço e distância para deslocamento de um docente de uma sala de aula para outra. Todas as salas de aula estão disponíveis todos os dias em período. Em cada sala de aula deve haver uma aula. Cada docente não pode estar alocado, ao mesmo tempo, em mais de uma aula.

A fig. 1 apresenta o algoritmo de Satisfação de Restrições implementado para o problema analisado. O algoritmo baseia-se na função heurística representada pela fórmula 7.

$$F = \min \sum_{i=1}^{m} restrições_professor(i) \qquad (6)$$
$$onde\ m = número\ de\ professores$$

O sistema é modelado em três fases distintas. A primeira fase, denominada Inicialização, compreende a definição das restrições individuais de todos os professores, para cada um dos vinte

horários semanais. Para cada horário, o professor deve escolher uma das quatro restrições possíveis: 0 (nenhuma), 1(restrição fraca), 2(restrição média) e 3(restrição forte). A restrição 0 indica que o professor pode lecionar naquele horário específico sem problema algum. A restrição 1 indica que o professor não gostaria de dar aula naquele horário, mas não haveria problemas maiores se ele fosse alocado. A restrição 2 indica que o professor teria sérios problemas para dar aula naquele horário, sendo que este deveria ser evitado a todo custo mas, em último caso, pode-se alocar o professor neste horário específico. Finalmente, a restrição 3 indica que o professor não pode, em hipótese nenhuma, lecionar naquele horário.

```
Fase de Inicialização:
Estabelecer Restrições dos Professores
  Cada professor deve indicar, para o seu
  horário semanal, um nível de restrição,
  utilizando a seguinte convenção:
    Restrição Forte - 3
    Restrição Média - 2
    Restrição Fraca - 1
    Sem Restrição - 0

Fase de Construção:
 Início:
 Fase = 3
 Enquanto Fase > 0
   Se Fase = 3 Entao Restricao = 3
   Senao Se Fase = 2 Entao Restricao = 2
   Senao Restricao = 1

   Para Todos os Professores
     Organizar os Professores em Ordem
       Decrescente do Somatório das suas
       Respectivas Restrições
     Se Professor(i).RestricaoHorario(t) > 0
       E Professor(i).RestricaoHorario(t) < Fase
       Entao
         Professor(i).RestricaoHorario(t) =
          Restricao

   Para Todos os Professores Organizados
     Para Todos os Horarios
       Se Professor(i).RestricaoHorario(t) < 3
       Então
         AlocarProfessor(i) em Horário(t)

   Se Não alocou todos os professores
     Fase = Fase - 1
     Va Para Início:

  Fim-Enquanto

Fase de Validação:
 Se, após as três fases, não houver um
 horário satisfatório, a tabela de restrições
```

Fig. 1. Algoritmo de Restrições

A segunda fase, denominada Construção, é formada pela utilização do algoritmo de satisfação de restrições para que este encontre um horário que satisfaça as restrições impostas pelo sistema e pelos professores. O algoritmo baseia-se numa estrutura de três fases. Na primeira fase, todas as restrições dos professores que forem maiores que zero (1, 2 ou 3) são consideradas como de nível 3. Esta característica influencia o processo no sentido de diminuir o número de disciplinas alocadas em horários com restrições. É importante salientar que um horário definido como de nível 3 não é alocado pelo algoritmo.

Na segunda fase, todas as restrições são consideradas como de nível 2, exceto aquelas que forem de nível 3. Na terceira e última fase, o nível de restrições é o mesmo definido pelos professores do curso.

Finalmente, após a construção do horário pelo algoritmo, a terceira e última fase do sistema, denominada Validação, é realizada pelos usuários encarregadas da construção do horário. Estes devem avaliar o resultado obtido através de critérios pré-definidos anteriormente, dentre os quais os mais comuns são o número de restrições aceitas e a factibilidade do horário gerado.

Se um horário não for gerado em sua completeza, ou seja, se o sistema não conseguir gerar um horário factível, deve-se modificar as tabelas de restrições, em comum acordo com os professores, pois elas provavelmente estão impedindo o sistema de obter um resultado satisfatório. Neste caso, o sistema retorna para a primeira fase, onde deve ser reinicializado.

V. CONCLUSÕES E TRABALHOS FUTUROS

Ao final dos estudos e testes realizados, pode-se constatar que a principal dificuldade é o tratamento do algoritmo quando há um grande número de restrições fortes a serem tratadas pelo problema, o que pode prejudicar o desempenho do mesmo.

Outra dificuldade encontrada foi o fato do sistema só estar aceitando, inicialmente, disciplinas apenas com quatro créditos. Logo, é necessário fazer alterações nas disciplinas com menores

créditos, para poder executar os testes no modelo proposto.

No entanto, para a elaboração de horário final, considerando apenas disciplinas com quatro créditos, o sistema teve um bom desempenho geral, gerando horários factíveis que poderiam ser armazenados e posteriormente utilizados como base para a elaboração de um horário mais otimizado. Logo, pode-se considerar que, como um modelo para a construção de horários, o sistema pode ser utilizado para a geração de uma solução inicial boa, onde os usuários ou um outro algoritmo de otimização pode lapidá-lo de acordo com as necessidades do problema.

Um problema crucial na utilização de um sistema automático para a geração de horários está na definição das restrições pelos professores. Usualmente, quando os professores tem total controle sobre quais dias eles querem ou não dar aula, os docentes acabam por cadastrar mais restrições que as necessárias, para tentar satisfazer critérios mais pessoais que profissionais. Alguns problemas comuns envolvem o fato de que a maioria dos professores preferem não ministrar aula na sexta-feira ou na segunda-feira, devido a proximidade do final de semana. Este problema, de origem cultural, pode acabar influenciando de forma decisiva na obtenção de resultados práticos. Uma forma de solucionar tal problema deve passar pela explicação da forma como o sistema funciona para todos os professores do curso e uma conscientização destes em relação ao cadastramento de restrição que não sejam necessárias.

Finalmente pode-se concluir que o modelo proposto no desenvolvimento da solução desta classe de problemas oferece praticidade, diminuição de esforço físico e tempo dispensado para elaboração de horários para uma Instituição de ensino, tendo vantagens, como a geração de vários resultados, tendo para isso a possibilidade de alteração da restrição dos professores.

Como trabalhos futuros, pode-se salientar a implementação da alocação de disciplinas com qualquer número de créditos e a utilização de uma heurística de melhoria, possivelmente um Algoritmos Genético, que busque uma solução ótima através da recombinação do conjunto dos melhores horários factíveis gerados até então, gerando um novo conjunto de horário para análise.

REFERÊNCIAS

[1] H.H. Weber. *Introdução à Pesquisa Operacional*. Ed. Universitária – UFPB. 1979.

[2] J.H. Osman. *Heuristcs for Combinatorial Optimization Problems: Developments and New Directions.* 1991.

[3] E. Rich., K. Knight. *Inteligência Artificial*. São Paulo:Makron Books, 1993.

[4] M.C. Goldbarg., H.P.L. Luna. *Otimização Combinatória e programação linear: modelos e algoritmos*. Rio de Janeiro : Campus, 2000.