

Ferramenta de Avaliação Automática no iGeom

Seiji Isotani^{1*}, Leônidas de Oliveira Brandão¹

¹Instituto de Matemática e Estatística – Universidade de São Paulo (IME-USP)
Caixa Posta 66.281 – 05315-970 São Paulo, SP

isotani@ime.usp.br, leo@ime.usp.br

Resumo. *Uma questão chave no aprendizado de matemática é a resolução de exercícios, entretanto, quando estamos considerando grandes turmas de alunos, a ausência de recursos que auxiliem o professor na sua correção, pode inviabilizar a aplicação de exercícios. Neste artigo apresentamos uma nova funcionalidade do programa iGeom, que resolve o problema de geração e correção automática de exercícios de Geometria. A solução implementada permite sua aplicação em aulas presenciais ou à distância, sendo que nesta última, devido ao mecanismo de comunicação do iGeom, pode ser empregada tanto em caráter aberto em páginas acessíveis por qualquer “internauta”, quanto em caráter fechado utilizando um sistema de controle de acesso.*

1. Introdução

O processo de aprendizado pode ser mais efetivo quando o aluno procura resolver, ele próprio, problemas adequadamente propostos. Como bem observa George Pólya [Pólya, 1975]: *A Matemática é a arte de resolver problemas ... e para resolver problemas é preciso resolver problemas.* Além deste, outro ponto importante no processo de aprendizagem é fornecer rapidamente ao aluno uma avaliação dos exercícios feitos. Entretanto, se o professor não dispuser de recursos auxiliares, precisará dispor de muito tempo para atender a estas demandas. Estas demandas, que valem tanto para o ensino presencial quanto à distância, podem ser atendidas por sistemas computacionais. Uma das formas mais antigas de atendimento são os sistemas com questões do tipo múltipla-escolha, que admitem avaliação rápida e simples, mediante gabarito [Gibson et al., 1995, Scapin, 1997, Abrão et al., 2004].

A questão da rapidez na apresentação da avaliação do aluno também é importante na aprendizagem, seja no modo presencial ou a distância. Como observam [Hara and Kling, 1999, Kirby, 1999], a principal frustração do aluno nos cursos de educação a distância é a limitação ou a falta de uma avaliação (*feedback*) imediata. Essa situação é crítica quando analisamos cursos com um grande número de alunos, como aponta [Bevilacqua, 2002] na disciplina Cálculo Numérico, oferecida para aproximadamente mil alunos dos cursos de Engenharia da Escola Politécnica da Universidade de São Paulo. Neste artigo apresentamos uma solução (sistema computacional) para ambas as pontas do problema de automatização de exercícios: a geração/publicação/avaliação por parte do professor e a resolução/verificação por parte dos alunos. O contexto considerado é o da Geometria e os exercícios são do tipo aberto (não múltipla-escolha).

A base do sistema é a Geometria Dinâmica (GD), que resumidamente pode ser entendida como a implementação computacional da “geometria tradicional”, aquela da régua-e-compasso. O termo “dinâmico” do nome pode ser melhor entendido como oposição à estrutura “estática” das construções da geometria tradicional. Na GD, após o aluno realizar uma construção, ele pode alterar as posições dos objetos iniciais e o programa redesenha toda sua construção, preservando as propriedades originais. Por exemplo, ao construir uma reta mediatriz m de dois pontos dados, A e B , pode-se mover A ou B livremente e o programa redesenha m para a nova

*Este autor contou com o apoio do CNPq.

configuração, aparentando um movimento contínuo. Em função desta possibilidade de alterar objetos preservando-se a construção, podemos dizer que a GD é uma geometria do tipo 1-construção, N-testes, enquanto a tradicional de régua e compasso é do tipo 1-construção, 1-teste [Brandão, 2004]. Esta é, para nós, a grande vantagem da GD sobre a geometria tradicional, pois permite que o aluno teste conjecturas e procure descobrir propriedades.

A GD começou a ganhar destaque na década de 90 [Botana and Valcarce, 2002], principalmente com a popularização dos programas comerciais *Cabri Geometry* [Laborde and Bellemain, 1997] e *Geometer's Sketchpad* [Jackiw, 1995]. Outros programas de GD são o *Cinderella* [Richter-Gebert and Kortenkamp, 1999], comercial, e *C.a.R.* [Grothman, 1999], gratuito. Estes dois últimos dispõem de um recurso de geração/avaliação de exercícios, sendo implementados em Java e por isso multi-plataforma. O sistema que estamos desenvolvendo, o *iGeom*, tem várias funcionalidades não encontradas simultaneamente nos outros programas de GD, como “scripts” recorrentes [Brandão, 2002], geração/avaliação de exercícios e recursos de comunicação. O *iGeom* é implementado em Java, é gratuito, e pode ser descarregado a partir do endereço <http://www.matematica.br/igeom>.

Este artigo está estruturado da seguinte forma, na seção 2 apresentamos o problema considerado, na seção 3 apresentamos o sistema de GD utilizado, na seção 4 definimos o critério de distância entre objetos, na seção 5 explicitaremos o algoritmo utilizado na avaliação e na seção 6, apresentamos os resultados obtidos experimentalmente.

2. O Problema da Avaliação

O propósito do trabalho aqui relatado foi desenvolver um recurso adicional no *iGeom* que permitisse ao professor produzir facilmente exercícios a serem resolvidos por alunos em qualquer navegador (com interpretador Java) e que, automaticamente, corrigisse a solução do aluno.

Resumidamente, estávamos interessados em uma implementação que atingisse os seguintes objetivos:

1. Detectar como corretas quaisquer soluções, mesmo que diferentes daquelas imaginadas pelo professor;
2. Apresentar de imediato o resultado da avaliação;
3. Fornecer um contra-exemplo se a resposta do aluno estiver errada.

Uma primeira questão que precisou ser considerada foi a possibilidade de um aluno resolver o problema de maneira muito diferente da imaginada pelo professor, eventualmente utilizando técnicas diferentes. A solução que adotamos é fortemente baseada na estrutura “dinâmica” da GD e parte de um gabarito que o professor deve fornecer, na geração do exercício. O aluno recebe este gabarito, mas só pode visualizar os objetos iniciais que descrevem o problema e não a resposta do professor. A avaliação é efetuada comparando-se os objetos-resposta do aluno com os correspondentes do gabarito do professor. Esta comparação é feita a partir de um critério de distância entre os objetos. Por exemplo, se o problema é determinar o ponto médio entre os pontos A e B , e o objeto-resposta do aluno tem o rótulo Ma e o do professor tem rótulo Mp , verifica-se qual a distância entre Ma e Mp , de acordo com o critério estabelecido (vide subseção 3.1).

Para evitar (no mínimo, reduzir) problemas de soluções que funcionem apenas em casos particulares, o processo de avaliação utiliza a estrutura “dinâmica” da GD: move-se os objetos iniciais (A e B no exemplo do ponto médio) e, para cada configuração, computam-se as distâncias entre as respostas do aluno e do professor. O resultado da avaliação é um algoritmo definido sobre estes resultados parciais.

Existem outras linhas de pesquisa que poderiam ser empregadas na avaliação de exercícios, como a *prova de teoremas* (*theorem proving*). Um trabalho pioneiro de prova de teoremas

em geometria euclidiana é o de [Gelernter et al., 1963, Gelernter, 1963] e outros mais recentes são os trabalhos de [Botana and Valcarce, 2002, Matsuda and Vanlehn, 2004].

3. O iGeom e a Avaliação Automática

O **iGeom: Geometria Interativa na Internet**, é um programa de Geometria Dinâmica que começou a ser desenvolvido em 2000, a partir dois projetos de iniciação científica no *Instituto de Matemática e Estatística da Universidade de São Paulo* (IME-USP), com os alunos Ricardo Hideo Sahara e Fabiana Piesigilli, ambos orientados pelo segundo autor. A segunda aluna, foi também orientada pelo professor Eduardo Toledo Santos da POLI-USP [Valente, 2003].

A interface do programa *iGeom* foi desenvolvida visando sua utilização tanto nas formas aplicativo, quanto *applet*. Além disso, foi utilizada uma versão “básica” do *Java* (AWT), com botões, para que pudesse ser utilizado em qualquer versão do *Java* a partir da versão 1.1. A menos das opções de gravação, que não são permitidas em *applet* (portanto via Internet) por razões de segurança, todas as demais estão acessíveis a partir destes botões.

A versão atual deste programa permite realizar todas as operações básicas de Geometria Dinâmica, como por exemplo: criar objetos geométricos (como pontos, retas, semi-retas, segmentos e circunferências); opções de edição (esconder/mostrar, remover ou desfazer remoção, rastrear objetos,...); e opções de gravação/recuperação de arquivos em diferentes formatos (incluindo gerar imagens em PostScript e GIF). Além das características usuais em programas de GD, o *iGeom* permite a geração de “scripts” (ou “macros”) recorrentes¹ e, desde o início de 2004, permite a autoria e avaliação de exercícios e, ainda, a comunicação visando seu emprego em sistemas de gerenciamento de aprendizado pela Web (com cadastro de alunos).

3.1. Autoria de Exercícios

Uma das características mais importantes na construção de sistemas de apoio ao ensino está no desenvolvimento de conteúdo, pois a dificuldade de se criar o conteúdo pode ser o fator determinante na sua utilização ou não, devido ao tempo necessário para a aprendizagem do sistema e para produção do material.

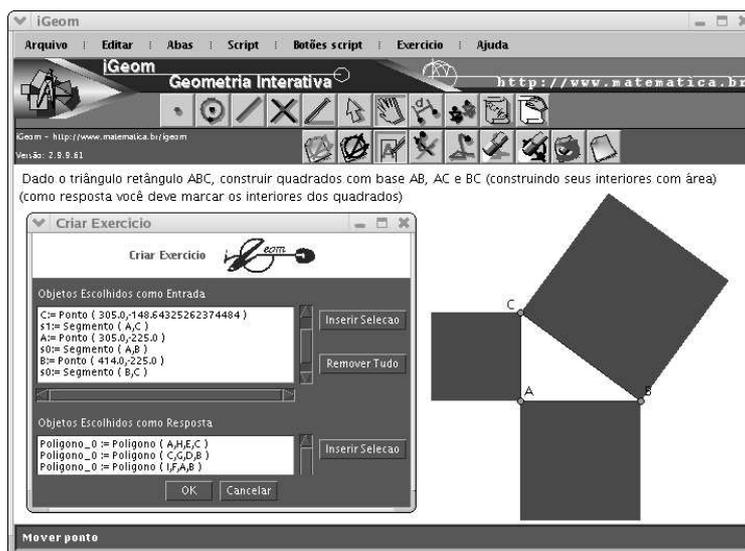


Figura 1: Janela para criação de exercícios no *iGeom*.

¹Dentre os outros programas de GD conhecidos, apenas o *Geometer's SketchPad* permite recorrências em “scripts”.

O processo de autoria de exercício no *iGeom* possui quatro etapas: construção da solução, seleção dos objetos resposta, seleção dos objetos de entrada e gravação do exercício. A interface de autoria do exercício é bastante simples, contando com uma janela para separar os objetos de entrada e de saída, dentre aqueles construídos na área de desenho (figura 1). A construção da solução de um exercício utiliza a mesma área de desenho que será utilizada em sua resolução. Dessa forma, o professor pode criar um exercício construindo os objetos nas posições onde gostaria que um aluno os visualizasse, agilizando sua formatação e publicação.

Quando a construção já estiver finalizada, é possível definir os objetos-resposta selecionando-os na área de desenho do *iGeom* e clicando no botão da janela que separa os objetos de resposta e de entrada (o mesmo procedimento é feito para definir os objetos de entrada). Após escolher os objetos de entrada e de resposta, o *iGeom* criará o exercício escondendo todos os objetos que não foram selecionados como entrada. Com o exercício criado, basta gravá-lo em arquivo para que qualquer usuário possa utilizá-lo ou exportá-lo para página Web.

Em modo *applet*, a construção de exercícios permite ao professor selecionar os recursos disponíveis (botões visíveis) aos alunos. Por exemplo, no problema do ponto médio o professor poderá desabilitar o botão de “ponto médio” existente no *iGeom* (figuras 2 e 3). Além disso, o professor pode decidir se o aluno receberá ou não uma mensagem mostrando o resultado da avaliação de sua construção.



Figura 2: Opções de ponto.



Figura 3: Opções de ponto sem o botão de “ponto médio”.

3.2. A Avaliação

Para que o processo de avaliação ocorra é necessário que o aluno use um arquivo gerado pelo professor (apresentado na seção 3.1), faça sua construção a partir desta base e anote quais são seus objetos-resposta. Após selecionar os objetos-resposta, o aluno pressiona o botão correspondente à “avaliação/envio de resposta” e o algoritmo de avaliação automática será iniciado. Caso o aluno selecione algum objeto de família geométrica diferente daquelas existentes no gabarito, ou o número de objetos-resposta do gabarito e do aluno não coincidirem, o aluno receberá uma mensagem de erro (seleção incorreta de objetos). Caso isso não ocorra, o algoritmo será disparado retornando: correto, correto com restrições (correto na maioria das instâncias), incorreto com restrições (incorreto na maioria das instâncias) ou incorreto.

Estes resultados poderão, ou não, serem apresentados ao usuário para notificá-lo do resultado da avaliação de seu exercício.

3.3. Comunicação

Para permitir que o *iGeom* seja utilizado em cursos “fechados”, com turmas de alunos matriculados e com um servidor, ele dispõe de recursos de comunicação. A comunicação é feita através de uma conexão HTTP direta entre o *applet* e o servidor e a troca de mensagens é feita utilizando o método POST [W3C, 2004]. A escolha dos protocolos HTTP e POST é devido a sua popularidade e por estar disponível em qualquer plataforma.

Este recurso, em conjunto com os anteriormente apresentados, proporciona a um aluno, conectado ao servidor, descarregar um exercício criado pelo professor, resolvê-lo e solicitar sua avaliação. Ao fazer esta requisição o *iGeom* irá avaliar o exercício (localmente) e enviar o resultado ao servidor (figura 4). Dentre as vantagens dessa implementação podemos listar:

1. um professor pode criar/editar um exercício em sua própria máquina e depois enviá-lo para o servidor ou pode fazê-lo pela Internet através de um navegador com interpretador *Java*;

2. fazer a adaptação da interface (ativar/desativar botões);
3. corrigir automaticamente os exercícios, oferecendo contra-exemplos quando for detectado erro;
4. enviar ao servidor a avaliação e a solução do aluno (ou um contra-exemplo para a mesma).

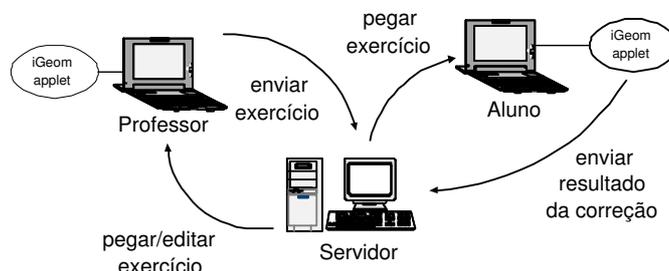


Figura 4: Comunicação entre um servidor e o *iGeom* em sistemas Web.

4. Avaliação Numérica

A linha de trabalho que adotamos é fortemente baseada no gabarito do professor, que deve ser suficientemente claro e não ambíguo. O resultado da avaliação é uma medida de distância entre a solução do aluno e o gabarito do professor. Como a distância é obtida a partir da descrição numérica dos objetos, denotaremos esta avaliação por **avaliação numérica**.

Para se fazer esta avaliação é necessário definir o critério de **distância** entre os pares de objetos geométricos. Neste artigo definimos o critério de distância apenas para pares de objetos de mesma família (ou tipo) do objeto. Para simplificar, vamos nomear apenas alguns dos exemplos de famílias mais comuns numa construção: família dos pontos (F_p); família das circunferências (F_c); e família dos segmentos (F_s). O conjunto de todas as famílias de objetos será representado por F_{og} .

Deste modo, o critério de distância pode ser uma função $dist$ que recebe um par de objetos geométricos $(og1, og2) \in F_{og} \times F_{og}$ e retorna um valor em \mathcal{R}_+ :

$$dist : (og1, og2) \longrightarrow \mathcal{R}_+. \quad (1)$$

A descrição computacional dos objetos, na maioria dos casos, é simples. Por exemplo: um ponto pode ser representado por um par (x, y) , onde x e y são as coordenadas do ponto; uma circunferência pode ser representada por uma tripla (x, y, r) , sendo (x, y) as coordenadas de seu centro e r seu raio; e um segmento $s = [(x_1, y_1), (x_2, y_2)]$, pode ser representado pela quádrupla (x_1, y_1, x_2, y_2) . Deste modo, se considerarmos apenas as famílias de pontos, circunferências e segmentos, podemos definir $dist$ conforme a equação 2. Seja $(l_1^i, l_2^i, \dots, l_k^i)$ a lista que descreve ogi , $i = 1, 2$, então

$$dist(og1, og2)^2 = \begin{cases} |l_1^1 - l_1^2| + |l_2^1 - l_2^2| & , (og1, og2) \in F_p \times F_p \\ |l_1^1 - l_1^2| + |l_2^1 - l_2^2| + |l_3^1 - l_3^2| & , (og1, og2) \in F_c \times F_c \\ \min \left\{ \begin{array}{l} \sum_{i=1}^4 |l_i^1 - l_i^2|, \\ \sum_{i=1}^4 |l_i^1 - l_{(i+1)\%4+1}^2| \end{array} \right\} & , (og1, og2) \in F_s \times F_s. \end{cases} \quad (2)$$

Uma vez definida a distância entre objetos, podemos definir a distância entre pares de construções distintas a partir de seus objetos. Sendo OG_p e OG_a , duas construções, e seus objetos

²O símbolo % está sendo empregado como o resto da divisão inteira (função módulo).

representados, respectivamente, por $(og_1^p, og_2^p \dots og_i^p)$ e $(og_1^a, og_2^a \dots og_j^a)$, a distância entre OG_p e OG_a pode ser expressa conforme a equação 3. Sendo $tipo(og)$ uma função que retorna o tipo do objeto geométrico og ,

$$dist(OG_p, OG_a) = \left\{ \begin{array}{ll} \sum_{i=1}^n dist(og_i^p - og_i^a) & , \#OG_p = \#OG_a = n \text{ e} \\ & tipo(og_i^p) = tipo(og_i^a), i = 1, \dots, n \\ +\infty & , \text{c.c.} \end{array} \right\}. \quad (3)$$

Esta é apenas uma das possibilidades para medir distância entre construções. Uma generalização simples desta função é colocar coeficientes positivos para ponderar cada objeto.

Uma solução (construção geométrica) pode ser representada como uma função que recebe uma lista de objetos geométricos (**entrada**) e retorna uma outra lista de objetos geométricos (**saída**), como na equação 4.

$$S : OG_i \longrightarrow OG_f. \quad (4)$$

Vale observar que a distância entre duas construções, efetuadas sobre o mesmo conjunto de entrada, pode depender de sua configuração inicial. Por exemplo, a partir dos pontos A e B pode-se, numa construção, produzir o ponto médio M de AB e noutra, produzir um ponto C tal que $dist(A, C) = dist(B, C) = dist(A, B)$ (ou seja, ABC é um triângulo equilátero). Neste caso, $dist(C, M)$ depende das coordenadas dos pontos A e B .

Uma **instância** de uma construção é uma construção aplicada sobre uma dada configuração de entrada.

Uma vez determinada a distância entre pares de listas de objetos e a representação de uma solução, podemos definir quando duas construções (soluções) são equivalentes, como segue. 4.1.

Definição 4.1 (Equivalência) *Sejam S_p e S_a duas construções aplicáveis sobre a mesma lista de objetos geométricos OG . Então S_a e S_p são equivalentes se, e somente se, para qualquer configuração OG_0 da lista OG , tivermos $dist(S_p(OG_0), S_a(OG_0)) = 0$.*

Vale observar que se duas construções são equivalentes, a distância entre ambas é invariante em relação às configurações iniciais, i.e., em qualquer instância a distância deve resultar 0. Este tipo de avaliação identifica como corretas quaisquer soluções, por mais diferentes que sejam. Entretanto existe o problema prático: como implementar uma versão suficientemente rápida e precisa ?

A saída que adotamos foi empregar o dinamismo da GD para verificar a distância em um número finito de instâncias. Esta solução, em princípio, permite que sejam construídos exemplos em que ocorram erros de *falso positivo* (exercício errado avaliado como correto) ou de *falso negativo* (exercício certo avaliado como incorreto), mas funciona bem na prática. Também é possível aumentar o número de instâncias a serem testadas ou fazer outras modificações de parâmetros para reduzir/eliminar avaliações falsas.

Quando estiver implícito, ou for indiferente, qual a lista de objetos geométricos OG , utilizaremos a notação simplificada S e não $S(OG)$.

5. O Algoritmo avaliador

Baseado na avaliação numérica apresentada na seção 4, implementamos no *iGeom* um algoritmo avaliador composto de quatro passos principais: instanciação, transformação numérica, avaliação e validação. Para efetuar a avaliação é necessário que o professor anote durante a construção do gabarito quais objetos serão avaliados, esta é a lista de **objetos-resposta** do professor. Durante a resolução do exercício pelo aluno, ele deve anotar quais são seus objetos-resposta, que devem ser

em número e tipos iguais ao do gabarito e depois solicita a avaliação³ da sua solução. As listas de objetos-resposta do professor e do aluno serão os dados de entrada do avaliador e o resultado será um natural entre 1 e 4, mas este intervalo de resultados é facilmente alterável. A execução deste algoritmo será apresentada nas duas sub-seções seguintes.

5.1. Transformação e Avaliação

A transformação numérica recebe um objeto geométrico e retorna uma lista de escalares que representam este objeto. Na maioria dos casos esta transformação é simples como apresentado na seção 4, porém alguns objetos, como o polígono, necessitam além da transformação numérica, a ordenação dos pontos que o representam. Com esta lista de escalares podemos fazer a comparação entre objetos. Assim, para analisar duas construções, o gabarito S_p do professor e a solução S_a do aluno, transformamos os objetos marcados como resposta em listas de escalares para então fazer a avaliação, como esquematizado na figura 5.

A avaliação é feita em duas etapas. A primeira etapa consiste no mapeamento entre as listas e a segunda na comparação delas através do critério de distância. A razão da primeira etapa é permitir que o aluno tenha a liberdade de fazer a marcação dos objetos-resposta em qualquer ordem.

Por exemplo, caso a solução do professor seja representada pelos pontos A e B e a do aluno pelos pontos C e D , a etapa de mapeamento identificará se o ponto C corresponde a A ou a B , fazendo o mesmo para D .

O mapeamento⁴ dos objetos de S_a e S_p é realizado comparando-se cada elemento og de S_a com todos os elementos de S_p que pertençam a mesma família de og e minimizando a distância entre eles. Assim, um objeto de S_a será mapeado em um objeto de S_p se ambos pertencerem à mesma família de objetos geométricos e a distância entre eles for a menor possível em relação aos outros objetos de S_a . Este mapeamento é feito apenas para a primeira instância (a configuração inicial do enunciado do exercício).

A segunda etapa da avaliação consiste na comparação entre os pares de objetos geométricos mapeados (og_i^a, og_i^p). Nesta comparação, utilizamos uma versão relaxada do conceito de equivalência expresso na definição 4.1, para levar em consideração as imprecisões numéricas ao se empregar diferentes soluções. Assim, adotamos uma margem de erro ε que permite descon siderar os erros numéricos gerados por imprecisões.

Definição 5.1 (Quase Equivalência) Fixado um conjunto de entradas OG , seja S_p uma construção sobre OG e S_a outra construção sobre o mesmo OG . Então S_a e S_p são quase equivalentes se, e somente se, para qualquer configuração OG_0 da lista OG , tivermos $dist(S_p(OG_0), S_a(OG_0)) < \varepsilon$.

Note que esta parte do algoritmo trata apenas uma **instância** da solução, considerando uma posição fixa para cada objeto de entrada. Se utilizarmos apenas esta avaliação da

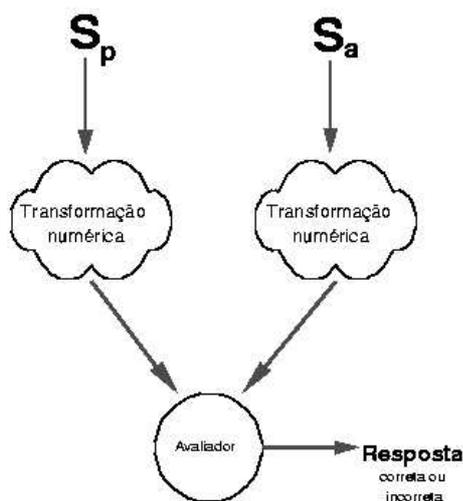


Figura 5: Transformação numérica e avaliação.

³No caso de uso do *iGeom* em um sistema fechado, com um servidor e alunos registrados, o professor pode optar por não mostrar ao aluno/internauta o resultado da avaliação e apenas arquivar no servidor a solução.

⁴Se os objetos coincidirem em número e tipo será uma função bijetora.

configuração inicial, pode ocorrer um erro de falso positivo. Por exemplo, no problema do ponto médio o aluno poderia tentar colocar um ponto “solto” sobre o segmento AB e movê-lo de modo a ficar próximo à posição do ponto médio, sem efetuar uma construção geométrica válida. Apesar de ser difícil posicionar o ponto de modo que o algoritmo avalie a solução como correta, é possível conseguir isso. A solução deste problema é apresentada na próxima subseção.

5.2. Instanciação e Validação

Uma maneira simples de detectar os erros apontados no final da seção 5.1 é criar um mecanismo que analise o exercício em várias instâncias (instanciação) e somente após um número considerável de avaliações o sistema retorna o resultado da avaliação (validação). A simulação deste procedimento pode ser visualizada na figura 6. A cada iteração do algoritmo de avaliação automática (figura 6) os objetos geométricos da entrada do exercício devem ter suas posições alteradas. Essa alteração é feita através da movimentação “aleatória” dos objetos, gerando diversas instâncias, para a verificação da quantidade de resultados considerados corretos e incorretos. Outros programas de geometria dinâmica como o C.a.R [Grothman, 1999] e o Cinderella [Richter-Gebert and Kortenkamp, 1999] possuem algoritmos semelhantes.

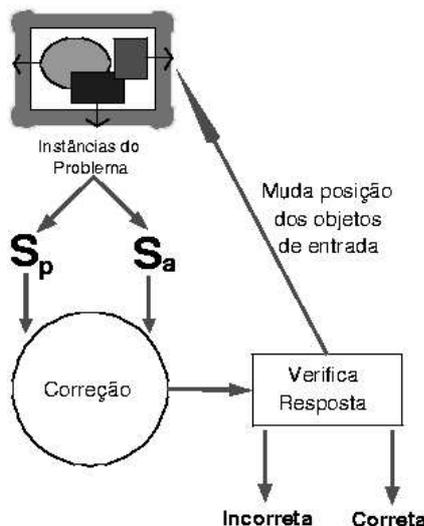


Figura 6: Avaliação Automática

6. Resultados Obtidos

Desde o início do primeiro semestre de 2004 está em teste um sistema de aprendizagem pela Web (SAW)⁵, com controle de acesso, utilizando o *iGeom*. Este está sendo desenvolvido em *PHP*, utilizando o gerenciador de banco de dados *MySQL*. O SAW está sendo utilizado por estudantes e professores em uma disciplina obrigatória oferecida para o curso de licenciatura em matemática do IME-USP, *Noções de Ensino de Matemática Usando Computador* (MAC118). Neste semestre ela foi ministrada em três turmas, uma diurna e duas noturnas, com 2 professores, 3 monitores e mais de 140 alunos.

Em edições anteriores de MAC118, todos os exercícios eram realizados utilizando o programa *iGeom*, mas sua correção era feita manualmente pelos monitores e professores. Devido ao número de alunos, a correção consumia grande parte do tempo dos monitores e o resultado da correção do exercício era entregue ao aluno duas ou três semanas após a realização do mesmo. Eram aplicados cerca de 20 exercícios por semestre. Com o desenvolvimento da ferramenta de correção automática no *iGeom*, além de reduzir o trabalho de professores e monitores, foi possível aplicar mais de 40 exercícios, desta vez, com a apresentação imediata da avaliação.

Através de dados obtidos no sistema SAW e do preenchimento de questionários pelos alunos, temos realizado análises para identificar dificuldades e obstáculos referentes ao uso destas ferramentas no ensino de geometria. Os resultados dessas análises servirão de base para aperfeiçoamento e criação de novas ferramentas no ambiente SAW e no programa *iGeom*.

Os primeiros resultados obtidos nos questionários são animadores. Através da aplicação da avaliação automática os alunos têm a possibilidade de tirar as dúvidas sobre a resolução do

⁵O Sistema de Aprendizado pela Web (SAW) faz parte do trabalho da mestranda Janine Gomes Moura.

exercício em sala de aula, como afirma um aluno: “... caso a construção estivesse certa, já estava enviada e caso estivesse errada, começaria novamente e tiraria as dúvidas em sala...”. Apesar de 97% dos alunos apoiarem o uso da ferramenta de correção automática, observamos que é necessário um cuidado maior em relação aos enunciados. Muitos deles tiveram dificuldades para resolver alguns exercícios (figura 7) devido à imprecisões em enunciado. As dificuldades por eles listadas, incluem: (a) a dificuldade de interpretar o enunciado; (b) a seleção incorreta dos objetos-resposta; e (c) a dificuldade de construir a solução.

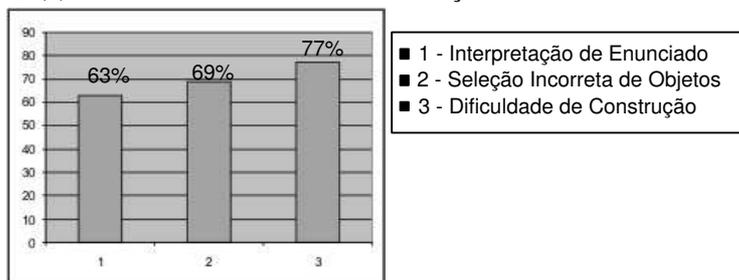


Figura 7: Dificuldades encontradas pelos alunos na tentativa de resolver os exercícios propostos.

7. Conclusões e Trabalhos Futuros

Este trabalho apresentou os desafios enfrentados e as soluções encontradas na implementação de autoria e avaliação de exercícios no *iGeom*. O algoritmo implementado já foi testado em 3 turmas, com mais de 140 alunos, durante o primeiro semestre de 2004, dentro de um sistema de aprendizado pela Web que está em fase inicial de desenvolvimento (SAW), na disciplina MAC118 (<http://www.ime.usp.br/~leo/mac118/04>). O resultado do trabalho pode ser conferido na atual versão do *iGeom*, disponível gratuitamente a partir do endereço <http://www.matematica.br/igeom>.

O recurso da autoria de exercícios facilitou a produção de exercícios por parte do professor e ainda pôde ser empregado em cursos fechados ou abertos na Web. Além disso, a possibilidade de rápida resposta ao aluno é um fator positivo à sua aprendizagem.

Através dos recursos apresentados, pretendemos estender o processo de interação dos alunos com os exercícios utilizando a animação. Dessa forma, será possível que o autor do exercício (professor) desenvolva animações (criação e movimentação automática de objetos geométricos na janela do *iGeom*) que utilizam as construções dos alunos como parte de uma animação, ou seja, a cada passo da animação será possível utilizar a resposta da construção do aluno para dar continuidade as animações seguintes. Utilizando o algoritmo avaliador será possível avaliar a resposta do aluno e, considerando-a correta a animação continuará normalmente, caso contrário a animação é paralisada para que o aluno possa refazer sua construção.

Além desta nova ferramenta, os desafios futuros que pretendemos vencer são: (a) a incorporação de recursos para melhorar a criação de exercícios e o *feedback* dado ao aluno; (b) o desenvolvimento de ferramentas para o compartilhamento das construções entre os alunos, viabilizando o trabalho cooperativo para solucionar um exercício proposto pelo professor; (c) e finalmente, pretendemos avaliar as ferramentas desenvolvidas para identificar as vantagens e desvantagens da utilização destas ferramentas no processo de ensino-aprendizagem.

Vale destacar que todas as ferramentas aqui discutidas poderão ser aplicadas em cursos presenciais ou a distância, permitindo que os professores possam criar e adaptar o material didático estático que normalmente é usado no ensino de geometria, trazendo benefícios ao aprendizado do aluno numa abordagem mais construtiva com exemplos interativos, atividades cooperativas e com avaliações rápidas e eficazes.

Referências

- Abrão, I. C., Rayel, F., and Abrão, M. A. V. L. (2004). Questcomp: Ferramenta para avaliação de aprendizado à distância. In *Proceedings of World Congress on Engineering Technology Education*, http://www.inf.pucpcaldas.br/rayel/download/papers/published/WCETE2004_Questcomp.pdf.
- Bevilacqua, J. S. (2002). Introdução da disciplina cálculo numérico no ambiente webct. In *Proceedings of VII International Conference on Engineering and Technology Education*, <http://www.asee.org/international/INTERTECH2002/566.pdf>.
- Botana, F. and Valcarce, J. L. (2002). A dynamic-symbolic interface for geometric theorem discovery. *Computer & Education*, 38(1–3):21–35.
- Brandão, L. O. (2004). Programação geométrica: Uso da geometria dinâmica para programação. In *Anais do Segundo Colóquio de História e Tecnologia no Ensino de Matemática*, pages 191–202.
- Gelernter, H. (1963). Realization of a geometry-theorem proving machine. pages 134–152.
- Gelernter, H., Hansen, J. R., and Loveland, D. W. (1963). Empirical explorations of the geometry-theorem proving machine. pages 153–163.
- Gibson, E. J., Brewer, P. W., Dholakia, A., Vouk, M. A., and Bitzer, D. L. (1995). A comparative analysis of web-based testing and evaluation systems. In *Proceedings of Fourth International World Wide Web Conference*, http://renoir.csc.ncsu.edu/Faculty/Vouk/Papers/Gibson/Gibson_WWW4_1995.pdf.
- Grothman, R. (1999). *C.A.R - Compass And Rules*. <http://mathsrv.ku-eichstaett.de/MGF/homes/grothmann/java/zirkel/>.
- Hara, N. and Kling, R. (1999). Student's frustrations with a web-based distance education course. *First Monday: Journal on the Internet*, 4(12). available at http://www.firstmonday.dk/issues/issue4_12/hara/.
- Jackiw, N. (1995). *The Geometer's Sketchpad v3.0*. Key Curriculum Press, Berkeley.
- Kirby, E. (1999). Student's frustrations with a web-based distance. In *Technology and Teacher Education Annual*, pages 199–205.
- Laborder, J. M. and Bellemain, F. (1997). *Cabri Geometry II*. Texas Instruments, Dallas.
- Matsuda, N. and Vanlehn, K. (2004). Gramy: A geometry theorem prover capable of construction. *Journal of Automated Reasoning*, 32(1):3–33.
- Pólya, G. (1975). *Arte de resolver problemas : um novo aspecto do método matemático*. Editora Interciência, Rio de Janeiro.
- Richter-Gebert, J. and Kortenkamp, U. (1999). *The Interactive Geometry Software Cinderella*. Springer-Verlag, Heidelberg.
- Scapin, R. H. (1997). Desenvolvimento de uma ferramenta para criação e correção automática de provas na world-wide-web. Dissertação de mestrado em física, Instituto de Física de São Carlos, Universidade de São Paulo.
- Valente, V. C. P. N. (2003). *Desenvolvimento de um ambiente computacional interativo e adaptativo para apoiar o aprendizado de geometria descritiva*. Tese de doutorado em engenharia, POLI - Universidade de São Paulo.
- W3C, W. W. W. C. (2004). *HyperText Markup Language (HTML)*. <http://www.w3.org/MarkUp/>.