

UNIVGO C'Y GD'RCTC'I GTCY'i Q'CWQO f VÆC'F Q'VGUVG'F G" RTQI TGUUQ'ð'HCUG'KK

Área temática: Metodologias e Abordagens de Ensino Aplicadas à Área de Ciência e Tecnologia

Gwi 'plq'Ukxc3=Lq q'Xlevqt 'f g'Uqw| c'I gqplk gnd4=IP dk 'Ngur 'Ecttgi cnEqt t 'c5=Gtle'Hcrē q'Tqej c6=

3locente do curso de graduação em Ciência da Computação UNIFESO;

4dicente do curso de graduação em Ciência da Computação UNIFESO;

5dicente do curso de graduação em Ciência da Computação UNIFESO;

6dicente do curso de graduação em Ciência da Computação UNIFESO;

PICPq 2022-2023.

TGUWO Q

Na Fase I deste trabalho propôs-se a implementação de um sistema web que oferecesse, a todos os atores envolvidos na elaboração do Teste de Progresso, um meio amigável e confiável para a geração da prova. Algumas das principais funcionalidades originalmente previstas para o sistema foram implementadas nessa fase, sendo a tarefa de implementação das funcionalidades restantes transferida para a Fase II, que se encontra em execução. Uma das funcionalidades de destaque do sistema é a geração automática da prova, cujo resultado deve ser um conjunto de questões que respeite ao máximo os parâmetros definidos pelo usuário e que estabelecem o perfil da prova. De acordo com essas características, o problema de geração da prova pode ser interpretado como um problema de otimização. Assim, neste trabalho é apresentada a modelagem do problema de otimização por trás da geração da prova usando a técnica de Algoritmos Genéticos. Nessa modelagem são apresentados a representação cromossômica a ser empregada para representar as potenciais soluções do problema, assim como os operadores genéticos a serem utilizados do processo evolutivo. Por fim, são definidos alguns aspectos importantes relacionados à função de aptidão que é responsável pela avaliação das soluções geradas durante o processo de otimização.

Palavras-chave: Teste de Progresso; otimização; algoritmos genéticos.

RPVTQFW'i Q

Desde 2009, quando passou a ser aplicado a todos os cursos de graduação do UNIFESO, o Teste de Progresso tem desempenhado um papel de destaque como um importante instrumento de avaliação institucional. Por meio dele, são obtidos indicadores que têm norteado tanto gestores acadêmicos quanto o corpo docente na tomada de decisões para o aprimoramento do processo de ensino-aprendizagem. Ademais, os próprios alunos também têm se beneficiado desse recurso como uma forma de medir o próprio crescimento acadêmico (MORGADO, 2014), (MIRANDA; MORAES, 2014).

A proposta de construção de um sistema web para a automatização da geração do Teste de Progresso foi originalmente apresentada no âmbito do Plano de Iniciação Científica e Pesquisa (PICPq), na edição de 2020-2021. Na ocasião propôs-se a construção de um ambiente que permitisse o trabalho colaborativo entre professores do UNIFESO, tanto para a elaboração das questões do Teste de Progresso, quanto para a geração da prova. A principal finalidade desse ambiente seria possibilitar a execução de todo o processo de construção da prova, desde a elaboração das questões, até a geração da prova, de uma forma mais amigável, confortável e confiável. Essa necessidade surgiu da percepção de que o processo tradicionalmente utilizado para a construção da prova era extremamente moroso e propenso a erros dos mais variados tipos (SILVA, GEONIZELI, FONTE, MELLO JÚNIOR, 2020).

Diante disso, foi proposta a construção de um sistema web que, de uma

perspectiva de mais alto nível, deveria contemplar as seguintes funcionalidades essenciais:

Ecf cwtq'f g's wguvz gu< permitir que professores cadastrados no sistema elaborem questões de forma a contemplar não apenas os seus elementos essenciais como, enunciado, suporte, resposta correta e distratores, mas também outros atributos importantes tanto para a parametrização quanto a geração da prova. Como parte desses atributos estão, por exemplo, o assunto, o tipo de questão, o grau de dificuldade e a habilidade cognitiva da Taxonomia de Bloom a ser avaliada pela questão.

Xcldf c± q'r qt'f ctgu< permitir, como parte do processo de criação de uma questão, que o elaborador indique o professor responsável pela sua revisão. Trata-se de uma funcionalidade de grande relevância, pois contribui para diminuir as chances de uma questão que contenha algum erro seja indevidamente incluída na prova. Com isso, evita-se a necessidade de eventuais anulações.

Rctco gvt k c± q'f c"rt qxc< permitir a configuração das características da prova, considerando percentuais associados aos atributos das questões, a fim de direcionar o processo de escolha das questões que vão compor a prova.

I gtc± q'f c'rt qxc< permitir a geração automática do arquivo da prova, com base nos parâmetros estabelecidos anteriormente para a seleção das questões.

Na edição 2020-2021 do PICPq foi possível concluir apenas o desenvolvimento das funcionalidades de cadastro de questões e validação por pares (SILVA, GEONIZELI, FONTE, MELLO JÚNIOR, 2020). Ainda nessa etapa, posteriormente denominada Fase I, foram acrescentadas duas novas funcionalidades que não estavam inicialmente previstas (SILVA, GEONIZELI, FONTE, MELLO JÚNIOR, 2021). São elas:

Gzldk± q'f g'wv wuf g's wguvz gu< apresentar ao usuário as questões existentes em seu perfil, bem como a situação de cada uma. Essas situações consideram as diversas etapas que compreendem o processo de elaboração de uma questão. Uma questão pode, por exemplo, estar em modo rascunho, ou esperando o parecer do revisor ou ainda pendente de alterações.

I gtc± q'f g' t' a hqu< exibir um painel com gráficos que apresentam a distribuição das questões registradas pelo usuário de acordo com diferentes aspectos. Dentre eles estão: assunto, tipo de questão, grau de dificuldade e habilidade do domínio cognitivo avaliado pela questão.

Na proposta submetida à edição 2022-2023 do PICPq, denominada Fase II, propôs-se a continuação do desenvolvimento do sistema, para que as funcionalidades de parametrização e de geração da prova pudessem ser contempladas. Assim como aconteceu na Fase I, na Fase II também foi percebida a necessidade de acrescentar mais uma funcionalidade de administração, descrita a seguir, que não estava inicialmente prevista:

Cf o lplmt c± q< permitir a inclusão e a exclusão de usuários do sistema e a configuração de seus perfis. Esses perfis possibilitam estabelecer os privilégios de visualização das questões cadastradas e também as permissões de execução de ações que envolvem, por exemplo, a criação de questões e a geração da prova. Além disso, permite ainda que sejam efetuadas alterações nas listas de valores dos atributos assunto e eixo de formação, que são selecionados durante a criação de uma questão.

O desenvolvimento da funcionalidade de administração já foi concluído (SILVA, GEONIZELI, MACHADO, RUFINO, 2022), restando ainda a implementação das funcionalidades de parametrização e de geração da prova. Neste trabalho, as atenções estiveram direcionadas à modelagem do problema de otimização que está por trás do processo de geração automática da prova.

LWUVHIE CVKXC

Conforme descrito em Silva, Geonizeli, Machado e Rufino (2022), logo após a aprovação da continuidade do desenvolvimento do sistema proposto em Silva, Geonizeli, Fonte e Mello Júnior (2020), denominada Fase II, no âmbito da edição 2022-2023 do PICPq, o UNIFESO adquiriu o sistema Qstione . O Qstione é uma plataforma comercial com funcionalidades semelhantes às do sistema originalmente proposto e, desde a sua aquisição, vem sendo usado para a construção de bancos de questões e para a elaboração das avaliações de disciplinas ligadas aos cursos de graduação da instituição. Além disso, em suas duas últimas edições, o Qstione foi utilizado também para a geração das provas do Teste de Progresso.

Apesar de a instituição já ter à sua disposição uma ferramenta para a elaboração das provas do Teste de Progresso, a continuidade do desenvolvimento do sistema originalmente proposto em Silva, Geonizeli, Fonte e Mello Júnior (2020) se justifica devido aos diferenciais apresentados em relação ao Qstione. Um desses diferenciais diz respeito à validação por pares. O sistema proposto permite sucessivas interações entre o elaborador e o revisor da questão, configurando assim um processo bidirecional de validação. O envio de réplicas, questionamentos ou esclarecimentos de parte a parte é permitido até que um consenso seja alcançado e a questão seja finalmente aprovada. Além disso, a escolha do revisor é uma atribuição do elaborador da questão, dando a ele a oportunidade de escolher um par que tenha condições de avaliar a questão tanto do ponto de vista da forma quanto do conteúdo.

No caso do Qstione, o processo de validação é unidirecional, uma vez que um parecer emitido pelo revisor não pode ser questionado pelo elaborador da questão. Além disso, no Qstione o elaborador da questão não tem qualquer controle sobre a escolha de seu revisor. Assim, não é possível garantir que o revisor seja alguém com conhecimentos aderentes ao tema da questão em julgamento. Com isso, perde-se a oportunidade de submeter a questão a uma avaliação mais ampla, que não se limite apenas ao formato, mas que se estenda também ao conteúdo. Outro diferencial de destaque do sistema proposto está na forma como a prova deve ser gerada. Além da forma manual, em que o usuário tem a liberdade para escolher as questões que vão compor a prova, segundo um conjunto de parâmetros, há também a forma automática, que prevê o uso de um modelo de otimização para montar a prova com um conjunto de questões que satisfaçam ao máximo a parametrização estabelecida pelo usuário.

QDLGVKXQU

Qdlgvkq'l gt cn

O objetivo geral do projeto ao qual este trabalho se associa, consiste em desenvolver um sistema web que permita que professores do UNIFESO trabalhem de forma colaborativa para a elaboração da prova do Teste de Progresso, de forma a tornar esse processo mais amigável e confiável.

Qdlgvkq'gur ge qleq

O objetivo específico desta etapa do projeto consistiu em modelar o problema de otimização subjacente à funcionalidade de geração automática da prova do Teste de Progresso. Para isso, o sistema deve oferecer um conjunto de filtros que permita ao usuário parametrizar a prova de acordo com as características desejadas. Um modelo de otimização baseado em um algoritmo genético deve se encarregar de selecionar as questões de forma a construir uma prova que respeite o máximo possível a parametrização desejada.

O GVQF QNQI KC

Rt qdigo cuf g'qvo k c± q

A geração da prova, funcionalidade considerada de destaque no sistema em desenvolvimento, pode ser interpretada como um problema de otimização. De um modo geral, um problema de

otimização se caracteriza por admitir uma grande quantidade de soluções e nesse universo de possíveis soluções deseja-se encontrar a melhor de todas. De maneira um pouco mais formal, um problema de otimização se caracteriza por uma busca que deve ser efetuada em um espaço de soluções de forma a encontrar aquela que maximize uma determinada métrica de satisfação.

Problemas de otimização são bastante comuns no cotidiano das pessoas e a complexidade por trás das soluções desses problemas está diretamente relacionada ao tamanho do espaço de busca e às eventuais restrições que devem ser respeitadas para que uma solução seja considerada viável. Assim, o mesmo problema de otimização pode ser considerado simples ou complexo dependendo da quantidade de possíveis soluções que precisam ser verificadas para que se encontre a melhor.

Um exemplo de problema de otimização clássico, bastante conhecido e que ilustra bem a diferença entre um problema simples e um problema complexo, é o Problema do Caixeiro Viajante (PCV). Trata-se de um problema teórico, em que se consideram um caixeiro viajante e um conjunto de cidades em que todas estão conectadas entre si, ou seja, de qualquer cidade, há sempre uma estrada que leva a outra. A cada conexão é atribuído um valor que representa a distância entre duas cidades. O objetivo do caixeiro é partir de uma cidade, visitar todas as outras cidades apenas uma vez e retornar à cidade de origem percorrendo a menor distância.

Em uma situação em que a busca pela melhor rota se resume a 4 cidades, o PCV se configura como um problema bastante simples. Afinal, para 4 cidades e considerando que a cidade de origem (e de término) é fixa, há apenas $3! / 2$ rotas diferentes a serem verificadas a fim de identificar a de menor distância. Nesse caso, mesmo um computador com baixo poder de processamento, usando uma estratégia de busca exaustiva, seria capaz de retornar a solução ótima em menos de um segundo. Por outro lado, se o número de cidades for aumentado para 50, a quantidade de rotas que devem ser verificadas é de $49! / 2$. Em um computador atual, de alto poder de processamento, a verificação de todas essas possibilidades demandaria um tempo superior a 1045 séculos, configurando-se assim como um problema intratável (ZIVIANI, 2011).

Apesar de ser um problema teórico, o PCV é um instrumento bastante útil para o desenvolvimento de estratégias para a solução de problemas de otimização reais. Isso porque há vários problemas reais que podem ser modelados como o PCV. Dentre eles estão: planejamento de viagens, roteamento de veículos, navegação de robôs e design de circuitos eletrônicos. Dessa forma, se um problema real pode ser modelado como o PCV, isso significa que as mesmas complexidades identificadas no PCV também estão presentes no problema em análise. Assim, se alguma estratégia de otimização se mostra eficiente para o PCV, provavelmente essa eficiência se estenderá aos problemas que se baseiam no PCV.

Para problemas complexos de otimização, está claro que estratégias exaustivas são inviáveis para a obtenção de soluções ótimas. Nesses casos, é necessário o emprego de estratégias que sejam capazes de efetuar uma busca mais criteriosa pelo espaço de soluções, de forma a obter, em um tempo aceitável, soluções que podem não ser garantidamente ótimas, mas que sejam minimamente razoáveis. Na impossibilidade de testar todas as soluções possíveis, o grande desafio enfrentado pelas diversas técnicas de otimização existentes consiste em estabelecer o melhor critério de exploração do espaço de busca, de forma a obter soluções melhores a um custo mais baixo.

No universo de métodos existentes que se dispõem a obter boas soluções para problemas complexos de otimização, estão aqueles denominados métodos heurísticos, que apresentam algumas características em comum. Dentre elas estão (1) a inspiração em algum comportamento observado na natureza; (2) a simplificação da modelagem do problema; e (3) a possibilidade de obter soluções satisfatórias, não necessariamente ótimas, em um tempo aceitável.

Historicamente, algumas das heurísticas que têm sido empregadas com sucesso em

muitos problemas complexos de otimização são: busca tabu, têmpera simulada, enxames de partículas, sistemas imunológicos artificiais e sistemas evolutivos. Os sistemas evolutivos, por sua vez, se desmembram em algoritmos genéticos, programação genética, programação evolutiva e estratégias evolutivas. Para este trabalho, são empregados os algoritmos genéticos como técnica para a construção do modelo de otimização para a geração da prova do Teste de Progresso.

Cri qtlwo qu'il gp² vlequ

Os algoritmos genéticos, propostos em Holland (1975), assim como outros métodos evolutivos, se baseiam em populações de potenciais soluções de um problema. Esse tipo de método se mostra bastante apropriado para a solução de problemas complexos de otimização, uma vez que, durante a evolução, são capazes de equilibrar adequadamente as capacidades de exploração do espaço de soluções e de aproveitamento das melhores soluções. Os métodos evolutivos, de maneira geral, se inspiram no Princípio da Seleção Natural, proposto por Charles Darwin em sua Teoria da Evolução das Espécies. De acordo com Darwin:

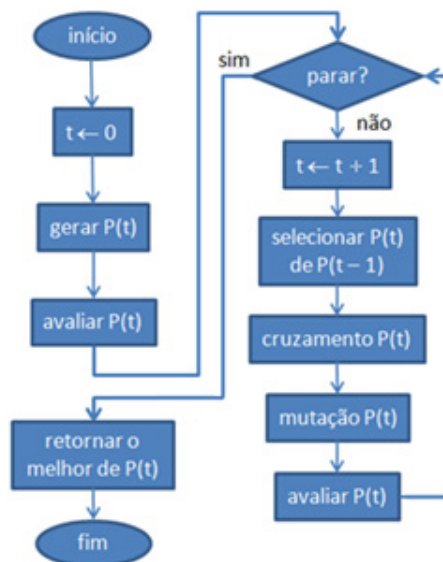
“Quanto melhor um indivíduo se adaptar ao seu meio ambiente, maior será sua chance de sobreviver e gerar descendentes.”

A partir desse princípio, a analogia que se estabelece é a de que cada solução possível do espaço de busca de um problema representa um indivíduo de uma espécie em um determinado ambiente. Naquele ambiente, esses indivíduos competem entre si e os mais bem adaptados ao ambiente tendem a prosperar e passar adiante o seu material genético, ao passo que os menos adaptados tendem a ser naturalmente eliminados. Trazendo para o universo dos problemas de otimização, espera-se que, ao submeter uma população de soluções a um processo evolutivo, as melhores prosperem em detrimento das piores.

Os problemas resolvidos por um algoritmo genético são do tipo $f:S \rightarrow R$, onde S é o espaço de busca composto por todas as soluções possíveis para um dado problema. A forma como as potenciais soluções para o problema é representada, varia de acordo com as suas particularidades, sendo a representação binária a mais comum. Nessa forma de representação, cada solução (indivíduo) é caracterizada por um cromossomo binário, que consiste em um vetor de tamanho n preenchido com valores 0 e 1. O valor de n é definido de forma a permitir que todas as soluções possíveis para o problema possam ser representadas por um número binário. Para cada cromossomo do domínio S , deve ser associado um valor de R que representa a aptidão do indivíduo representado por aquele cromossomo. De outro modo, essa aptidão representa uma medida de avaliação que quantifica o nível de adequação daquela solução ao problema em análise.

A principal tarefa de um algoritmo genético consiste em buscar, de forma eficiente, em amostras do espaço de busca S , soluções que satisfaçam o objetivo do problema. Vale destacar que, especialmente em problemas em que os espaços de busca são muito amplos e complexos, as soluções encontradas não são necessariamente ótimas, mas devem ser minimamente satisfatórias. O processo de evolução empreendido por um algoritmo genético é representado pelo fluxograma apresentado na Figura 1:

Figura 1: fluxograma de um algoritmo genético clássico



O primeiro passo consiste em gerar a população inicial de indivíduos (gerar $P(t)$) que, em geral são gerados de forma totalmente aleatória. Em seguida, calcula-se a aptidão (avaliar $P(t)$) de todos esses indivíduos. A partir daí inicia-se a evolução que se caracteriza pela execução de um conjunto de ações que se repetem até que uma condição de parada seja alcançada. A primeira dessas ações consiste na aplicação de um mecanismo de seleção (selecionar $P(t)$ de $P(t - 1)$) que simula o princípio de seleção natural proposto por Darwin. O objetivo aí consiste em privilegiar a preservação dos indivíduos mais aptos. Para isso, os indivíduos são distribuídos em uma roleta de maneira que os mais aptos ocupem porções maiores dessa roleta. Assim, ao ser girada, os indivíduos que ocupam porções maiores da roleta terão chances maiores de serem selecionados para compor a população da próxima geração.

Durante a evolução, o tamanho da população permanece constante e, por isso, na seleção a roleta deve ser girada uma quantidade de vezes igual ao tamanho da população. Vale destacar que o tamanho da população é um parâmetro que deve ser informado como entrada para o algoritmo. Devido às características da roleta, espera-se que a cada geração os indivíduos mais aptos se perpetuem e que os menos aptos sejam eliminados. A Figura 2 apresenta um exemplo de uma roleta com 5 indivíduos. Nela é possível observar que o indivíduo 1 é o de maior aptidão, uma vez que ocupa a maior porção da roleta. Por outro lado, o indivíduo 3 é o de menor aptidão. Portanto, há uma probabilidade bastante alta de que o indivíduo 1 seja preservado de uma geração para outra, enquanto o indivíduo 3 tende a ser eliminado.

Figura 2: roleta de seleção de indivíduos



Após a seleção, os operadores genéticos de cruzamento (cruzamento $P(t)$) e de mutação (mutação $P(t)$) efetuam modificações em alguns dos indivíduos da população a fim de gerar indivíduos novos. A cada um desses operadores é associada uma probabilidade de aplicação e a partir dela é que se estabelece o subconjunto da população a ser modificado. Importante destacar que, assim como o tamanho da população, as probabilidades de cruzamento e de mutação também são parâmetros fornecidos como

entrada para o algoritmo. A eficiência do processo de otimização por trás da evolução depende, dentre outras características, de um equilíbrio adequado entre as aplicações dos operadores de cruzamento e de mutação.

O cruzamento consiste em recombinar o material genético de dois indivíduos para criar dois novos, enquanto a mutação envolve a modificação de uma ou mais características de um indivíduo para transformá-lo em um novo. No caso do cruzamento, observa-se ainda que a recombinação envolve o secionamento dos cromossomos em um determinado ponto que, por sua vez, é determinado de forma aleatória. Tomando como exemplo um problema em que os indivíduos são representados por cromossomos com codificação binária, as Figuras 3 e 4 ilustram, respectivamente, o funcionamento dos operadores de cruzamento e de mutação:

Figura 3: operador clássico de cruzamento

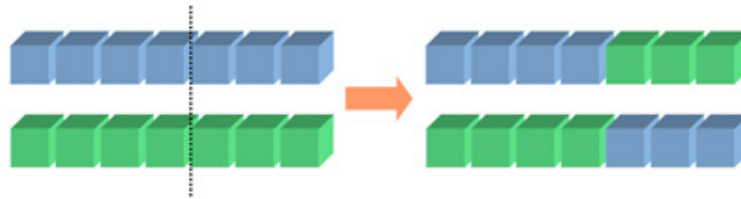


Figura 4: operador clássico de mutação



Após a aplicação dos operadores, tem-se uma nova população e, para encerrar o ciclo de evolução, é necessário calcular a aptidão dos indivíduos dessa nova população (avaliar $P(t)$). Esses ciclos se repetem até que a condição de parada seja alcançada. Em geral, essa condição de parada é dada pela quantidade de ciclos de evolução, que também é um parâmetro que deve ser informado como entrada para o algoritmo. Ao final, espera-se que, de acordo com o princípio de Darwin, a melhor solução obtida (retornar o melhor de $P(t)$) seja melhor que aquelas obtidas nas gerações anteriores.

Há alguns aprimoramentos que podem ser incluídos em um algoritmo genético para torná-lo ainda mais eficiente. Há ainda outras formas de representação cromossômica e de operadores genéticos que tornam o algoritmo genético mais aderente ao problema de otimização em estudo. Mais informações sobre aprimoramentos, formas de representação e operadores genéticos podem ser obtidas em Michalewicz (1996) e Linden (2008).

TGUWVCFQU'GFÆEWUÏ Q

Algoritmos genéticos são um tipo de técnica de otimização bastante versátil que se aplica a uma variedade enorme de problemas. O desafio a ser superado para o emprego dessa técnica em algum problema está na modelagem. No caso de um algoritmo genético, o processo de modelagem envolve, primeiramente, a definição da representação cromossômica de seus indivíduos, ou seja, a forma como as soluções serão representadas.

Uma vez encontrada a melhor forma de representação, o próximo passo consiste em definir os operadores genéticos de cruzamento e de mutação que sejam adequados ao tipo de representação cromossômica estabelecida. O ideal é que esses operadores sejam definidos de forma a garantir que ao efetuar alguma modificação em um cromossomo ou recombinação entre cromossomos, não sejam gerados cromossomos considerados inválidos, ou seja, cromossomos que representem soluções inviáveis para o problema em análise. Na impossibilidade de garantir que isso não aconteça, uma medida mitigatória consiste em aplicar algum tipo de penalidade a esses indivíduos que impactem em sua aptidão e, com isso, diminua a sua probabilidade de ser selecionado para compor uma próxima geração.

Por fim, ainda como parte do processo de modelagem, o outro passo consiste em definir a função

de aptidão. É por meio dessa função que o algoritmo atribui a cada indivíduo da população uma espécie de nota que permite avaliar quantitativamente cada indivíduo e, conseqüentemente, comparar os indivíduos

O código genético

Para o problema de otimização por trás da geração da prova do Teste de Progresso, a representação cromossômica a ser adotada consiste em um vetor com n números inteiros, onde n representa a quantidade de questões a serem inseridas na prova e cada número inteiro representa o identificador de alguma questão que esteja no banco de questões. Assim, o indivíduo representado pelo cromossomo consiste em uma possível configuração de questões para a prova. Um exemplo de cromossomo é apresentado na Figura 4:

Figura 4: exemplo de cromossomo que representa uma prova

45 32 17 28 95 74 23 15 94 67

No exemplo apresentado tem-se um vetor de 10 posições, representando uma prova com 10 questões. Em cada posição do vetor tem-se um valor correspondente ao identificador de uma questão que esteja armazenada no banco de questões do sistema. Para a representação cromossômica estabelecida, os operadores genéticos clássicos de cruzamento e de mutação se mostram apropriados e devem ser empregados. No caso do operador de cruzamento, a substituição da representação binária pela representação baseada em números inteiros não exige qualquer modificação na forma de implementação do operador. O critério de recombinação de dois cromossomos em algum ponto pode ser utilizado indistintamente em ambas as formas de representação.

No caso do operador de mutação, contudo, é necessário um pequeno ajuste na implementação. Enquanto na representação binária a mutação consiste em inverter alguns valores do cromossomo (trocar 0 por 1 ou trocar 1 por 0), na representação adotada para a geração da prova, a mutação consiste em substituir alguns valores do cromossomo por outros correspondentes a outras questões disponíveis no banco de questões do sistema.

Apesar de os operadores descritos serem considerados adequados à nova forma de representação cromossômica, não se pode garantir que os indivíduos gerados após a aplicação desses operadores sejam válidos. Isso porque as modificações empreendidas podem gerar cromossomos em que o mesmo código de questão apareça mais de uma vez no mesmo cromossomo. Na prática, isso significa ter uma prova com questões repetidas, o que não é permitido. Na falta de operadores mais eficientes, que evitem a geração de indivíduos inválidos, a alternativa a ser adotada consiste em penalizar, por meio da função de aptidão, os eventuais indivíduos inválidos gerados durante a evolução.

Por fim, a função de aptidão ser definida considera o problema de otimização como um problema de maximização. Isso significa que os indivíduos mais aptos, ou seja, com melhor avaliação, são aqueles que assumem os maiores valores de aptidão. Para a construção da função são levados em consideração todos os parâmetros que compõem as características da prova que se quer gerar. Esses parâmetros envolvem: intervalo para o ano de criação da questão; o fato de a questão ter sido usada ou não em outras provas; o tipo de questão objetiva; o grau de dificuldade; a habilidade cognitiva a ser avaliada; e o eixo de formação ao qual a questão pertence.

Para a geração da prova, o usuário deve ter a liberdade de selecionar um ou mais desses parâmetros para compor a função de aptidão. Dessa forma, a função de aptidão a ser criada deve ser uma expressão que retorne valores mais altos para configurações de prova que respeitem ao máximo os valores estabelecidos para os parâmetros escolhidos para compor a função de avaliação. Vale destacar que também deve ser incorporada a função de avaliação um fator que represente a penalização a ser aplicada a indivíduos inválidos. O trabalho de elaboração da função de aptidão, levando em conta todas as características apresentadas, ainda está em execução.

EQPUFGTCY ð GUHPCKU

Na Fase I do projeto ao qual este trabalho se associa (edição 2020-2021 do PICPq), propôs-se a implementação de um sistema web que pudesse tornar a tarefa de elaboração da prova do Teste de Progresso mais amigável, confortável e confiável. Nessa Fase I foi possível contemplar algumas das funcionalidades essenciais do sistema, como o cadastro de questões e a validação por pares, e também duas novas funcionalidades que não haviam sido originalmente previstas, a exibição de status de questões e a geração de gráficos.

A Fase II (edição 2022-2023 do PICPq) prevê a conclusão da implementação das outras funcionalidades previstas para o sistema como a parametrização e a geração da prova. Até o momento foi concluída a implementação da funcionalidade de administração que, apesar de também não ter sido prevista na proposta original, se mostrou de grande importância para a gerência do sistema.

Neste trabalho, especificamente, foi apresentada a modelagem do problema de otimização subjacente à funcionalidade de geração da prova, usando a técnica de Algoritmos Genéticos. Como parte dessa modelagem foram definidos a representação cromossômica dos indivíduos a serem submetidos ao processo evolutivo e também os operadores genéticos. A função de aptidão encontra-se ainda em elaboração, mas alguns de seus aspectos mais importantes já estão devidamente definidos.

Os próximos passos consistem, primeiramente, em concluir a elaboração da função de aptidão e em seguida implementar o modelo de otimização. A implementação desse modelo envolve também a implementação das funcionalidades de parametrização e geração da prova que ainda restam para a conclusão do projeto.

TGHGTŽ PEKCU

Holand, JH, *Adaptation in Natural and Artificial Systems*. Ann Arbor: University of Michigan, 1975.

Linden, R, *Algoritmos Genéticos: Uma Importante Ferramenta de Inteligência Computacional*, 2ª ed., Rio de Janeiro: Brasport, 2008.

Michalewicz, Z, *Genetic Algorithm + Data Structures = Evolution Programs*. Germany: Springer-Verlag, 1996.

Miranda, JFA, Moraes, MBVB. PAAI: Programa de Autoavaliação Institucional. Autoavaliação Institucional no UNIFESO – 15 Anos de Avaliação Transformadora. Vol. 1. Teresópolis: UNIFESO; 2014.

Morgado. FEF. O Teste de Progresso. Autoavaliação Institucional no UNIFESO – 15 Anos de Avaliação Transformadora. Vol. 1. Teresópolis: UNIFESO; 2014.

Silva, E, Geonizeli, JVS, Fonte, RC, Mello Júnior, LCR. Sistema Web para Geração Automática do Teste de Progresso. Anais do V Congresso Acadêmico Científico do UNIFESO – CONFESO, Teresópolis: Editora UNIFESO, 2020.

Silva, E, Geonizeli, JVS, Fonte, RC, Mello Júnior, LCR. Sistema Web para Geração Automática do Teste de Progresso. Anais do VI Congresso Acadêmico Científico do UNIFESO – CONFESO, Teresópolis: Editora UNIFESO, 2021.

Silva, E, Geonizeli, JVS, Machado, JVS, Rufino, JV. Sistema Web para Geração Automática do Teste de Progresso – Fase II. Anais do VII Congresso Acadêmico Científico do UNIFESO – CONFESO, Teresópolis: Editora UNIFESO, 2022.

Ziviani, N, *Projeto de Algoritmos - com Implementações em Pascal e C*, 3ª edição revista e ampliada, São Paulo: Cengage Learning, São Paulo, 2011.