

APLICAÇÃO DE ALGORITMOS GENÉTICOS E MÉTODOS EVOLUCIONÁRIOS NA RESOLUÇÃO DO PROBLEMA DE ALOCAÇÃO DE TURMAS: CASO UFPR.

Pablo Leonn Baptistello de Souza
Universidade Federal do Paraná - UFPR
pablobaptistello@gmail.com

Cassius Tadeu Scarpin
Universidade Federal do Paraná – UFPR
cassiusts@gmail.com

Resumo

Esse artigo apresenta uma adaptação do algoritmo genético na resolução do problema de alocação de turmas no campus Politécnico da Universidade Federal do Paraná. Para tanto, foram analisados diversos métodos, sendo o escolhido, devido sua eficiência e adaptabilidade, o algoritmo genético. O algoritmo desenvolvido, além de ser testado com dados reais e atuais da instituição, foi comparado com o processo atual aplicado no campus, obtendo resultados mais satisfatórios em relação ao mesmo. O sistema criado conseguiu alocar, através da inserção de novos conceitos e adaptações nos operadores genéticos do algoritmo genético clássico, 1348 turmas em 58 salas em diversos períodos e dias da semana de forma rápida e eficiente em função das restrições e particularidades do problema.

Palavras Chave: Algoritmo Genético; Grade Horária em Universidades; Meta-Heurística.

Abstract

This paper presents an adaptation of genetic algorithm in solving the problem of allocating classes at the Polytechnic campus of the Federal University of Paraná. Therefore, several methods were analyzed, wherein the chosen was, because of their efficiency and adaptability, the genetic algorithm. The developed algorithm, in addition to being tested with real and current data of the institution, was compared with the current process applied on the campus, obtaining more satisfactory results compared to the same. The created system managed to allocate, by the insertion of new concepts and adaptations of the genetic operators of the classic genetic algorithm, 1348 classes in 58 rooms at different times and days of the week quickly and efficiently in function of the restrictions and particularities of the problem.

Keywords: Genetic Algorithm; Timetabling in Universities; Metaheuristic.

1. INTRODUÇÃO

O problema apresentado neste artigo se enquadra em um amplo tema abordado na Pesquisa Operacional: o problema de designação. Neste tipo de situação, o objetivo é alocar os recursos a um determinado destino levando em consideração diversas restrições e particularidades de cada problema, como capacidade do destino, por exemplo. A solução buscada deve apresentar uma designação que vise atender de forma efetiva e factível as restrições impostas [3]. Devido ao grande número de variáveis relacionadas a esse tipo de situação e as singularidades das restrições, o método exato se torna demorado ou, em alguns casos, até mesmo inviável. Entretanto, métodos alternativos, heurísticos e meta-heurísticos, vêm sendo usados para encontrar uma solução próxima a solução ótima [1,3-7,10,13].

O problema abordado no estudo é a alocação de turmas, com diversas quantidades de alunos, em salas de aula, com tamanhos diferentes e quantidade de carteiras escolares diferentes, no campus Centro Politécnico da Universidade Federal do Paraná (UFPR). Este problema pode ser abordado como uma subcategoria do problema descrito. A problemática consiste em alocar um grande número de turmas a um limitado número de salas, levando em consideração a grade horária pré-definida, preferências das turmas em relação ao bloco/prédio de estudo e capacidade das salas (medida em número de alunos que a sala atende). O problema torna-se mais complexo devido ao número restrito de salas, sendo que, em determinados horários, não é possível alocar todas as turmas em salas disponíveis com tamanho adequado, forçando-as a serem designadas a outro campus. Sendo assim, a qualidade da solução, que será a designação de turmas grandes em salas grandes e turmas pequenas em salas pequenas, preferencialmente nos blocos adequados vinculados ao curso torna-se um fator essencial para o problema em questão, considerando que existem cursos que não possuem blocos específicos e que estes ocupam as salas nos blocos existentes de outros cursos.

Atualmente, o método utilizado para alocação das turmas é realizado por redes neurais artificiais (RNA). Contudo, os resultados obtidos vêm gerando conflitos de ensalamento que demandam correções manuais posteriores.

Buscando outra abordagem na solução desse problema, uma adaptação de métodos evolucionários e algoritmos genéticos foram desenvolvidos. O objetivo principal foi implementar um sistema que apresentasse como solução uma tabela de horários efetiva e viável, no qual a análise e processamento de informações ocorresse de forma generalizada (para a semana inteira) e adaptativa (gerando o menor número de designações erradas quanto ao tamanho e ao bloco). Para tanto, foram pesquisadas diversas metodologias usadas em problemas similares, sendo que a escolhida, devido a promissora qualidade de seus resultados e facilidade na adaptação das restrições [1,3-8,10-13], foi o tratamento evolucionário para o problema. O modelo desenvolvido foi testado em diversas situações para o caso da UFPR, e, em comparação com a solução atual, obteve resultados mais satisfatórios, minimizando conflitos e atendendo de forma efetiva as restrições do problema.

2. DESCRIÇÃO DO PROBLEMA

O estudo contempla a designação de salas para os 19 cursos de graduação que utilizam as salas do campus Centro Politécnico da UFPR, divididos em três setores, sendo eles: ciências exatas, ciências da terra e tecnologia. Esses cursos possuem aulas distribuídas entre segunda-feira e sábado, em diferentes horários, sendo que o primeiro período de aula começa às 07:30h e o último termina às 23:30h. Cada período tem duração de uma hora aula.

Para que fosse possível realizar uma comparação viável com o método atual utilizado, foram usados os mesmos dados utilizados no método RNA atual, salvo em relação aos laboratórios, ateliês e turmas que demandam esses tipos de sala, devido à falta de informações referentes a esses itens. Por exemplo, na contabilização dos dados, há, no período

do primeiro semestre de 2014, 1396 turmas e 58 salas disponíveis, sendo que estas estão distribuídas em 11 blocos dentro do campus Centro Politécnico.

Uma dificuldade do problema encontra-se na variabilidade do número de alunos em uma turma e a capacidade das salas disponíveis, sendo que esses valores podem variar de 10 a 110 alunos por turma e quantidade de carteiras em sala de aula. Além disso, há turmas que têm preferências por salas em relação ao seu tipo ou bloco situado, como, por exemplo, uma turma de Arquitetura e Urbanismo tem preferência por salas no bloco PD. Outra restrição relevante é o fato de que, preferencialmente, uma turma deve estar alocada a mesma sala em diferentes períodos da semana, visando facilitar a grade horária dos alunos, ou seja, uma turma que tenha aula ministrada no primeiro período de segunda-feira e quarta-feira, deve ser designada na mesma sala, nesses dois horários, se for viável. Ainda, para possibilitar a execução do algoritmo, as turmas e salas foram subdivididas em elementos que levam em consideração seus respectivos horários, sendo que em média, uma turma necessita de três períodos da semana e nesta há um total de 100 horários. Essa adaptação implica num grande número de dados de processamento, sendo eles: 2924 elementos “turma” e 5800 elementos “sala”. Um problema implícito nestes citados é o fato de que se uma turma tiver aula das 07:30h às 9:30h em um bloco situado em um extremo do campus e das 9:30h às 11:30 tiver aula em uma sala no outro extremo do campus, pode ocorrer atraso da segunda aula, devido ao tempo gasto no deslocamento dos alunos pelo campus.

Um facilitador em relação dos outros trabalhos correlatos [1,3,7,10] é o fato de que a grade de horários que relacionam os professores, as disciplinas e os horários das aulas, é previamente determinado, independente da execução do método, diminuindo assim a complexidade do problema, mas não retirando o mérito em sua realização.

Um ponto importante a ser ressaltado é que, mesmo que alguns dados relacionados as singularidades de sala (laboratórios, ateliês) e turma (turmas de desenho, computação) não foram consideração na comparação dos métodos, o método desenvolvido foi preparado para processar e compilar esse tipo de informação.

3. REVISÃO BIBLIOGRÁFICA

3.1. TRABALHOS CORRELATOS

Beligiannis *et al* [3] propôs um algoritmo evolucionário para a resolução do problema das grades horárias nas escolas da Grécia. No estudo, os autores aplicaram e testaram exaustivamente seu método em 30 escolas situadas na cidade de Patras, visando comprovar a excelência e flexibilidade da metodologia criada com dados reais. Ainda, para validar do estudo realizado, o algoritmo foi comparado com outros dois métodos que eram utilizadas para a formulação da tabela de horários no país, obtendo resultados mais relevantes em relação os anteriores.

Através de um da aplicação do algoritmo genético em duas fases, Mozos *et al* [10], pode solucionar de forma satisfatória o problema de formulação das grades horárias na Universidade de Vigo, na Espanha. O método proposto opera duas fases, na primeira, são alocados os estudantes que tem alguma preferência específica, e em subsequência, os estudantes sem preferências por recurso. Nesse enredo, o estudo realizado com um número de 1301 alunos, pode alocar de forma considerável os estudantes que não possuíam preferência no algoritmo, obtendo um resultado de 99% de designação com as turmas analisadas.

Colorni, *et al* [5] realizaram uma comparação entre a utilização das meta-heurísticas algoritmo genético, *simulated annealing* e busca tabu para o problema da programação de horários de escolas do ensino médio. Concluiu-se que, através da implementação do método a um problema específico, a busca tabu obteve os melhores resultados. Porém, destaca-se a superioridade do algoritmo genético pelo fato de permitir a uma adaptabilidade maior em relação aos recursos e restrições, fator fundamental devido à grande variabilidade no problema de designação de horários em relação a cada caso.

Um problema similar ao abordado neste trabalho foi tratado por Blas *et al* [4]. No trabalho realizado por estes, um algoritmo genético híbrido foi desenvolvido para a resolução do problema de alocação de turmas em laboratórios. O estudo analisa o uso do algoritmo genético e operações de agrupamento para a criação de um método híbrido, que atendesse de forma eficiente as particularidades do problema. O procedimento foi aplicado na Universidade de Alcalá, em Madri, alcançando resultados notáveis no tratamento do problema.

Na China, Duan *et al* [7], propõem uma nova abordagem visando otimizar o uso de algoritmos genéticos para o tratamento do problema de grades horárias em escolas. Os autores apresentam em seu estudo, através da análise dos fatores relacionados ao problema em questão, o uso da *fuzzy synthetic decision-making* para a otimização e criação de uma nova codificação para algoritmos genéticos. O resultado obtido apresentou uma rápida convergência para os ótimos locais, confirmando a viabilidade da metodologia apresentada.

3.2. ALGORITMO GENÉTICO

Um algoritmo genético (AG) consiste em um método probabilístico de otimização baseado em processos evolucionários como seleção natural e dinâmica genética. Os primeiros fundamentos descritos sobre este método foram apresentados por Holland em 1975, sendo amplamente difundidos, devido a adaptabilidade em sua aplicação e qualidade de seus resultados [2,9].

Um AG consiste em um processo de busca em populações, no qual os indivíduos inseridos nela sofrem a ação de operadores genéticos para aprimorar a função objetivo resultante em cada geração. De forma iterativa, esse processo de busca e otimização se repete até que o resultado se mostre adequado [2,9].

Em relação aos cromossomos, estes podem ser representados de diversas formas dependendo da situação, sendo a binária a mais comum delas. De acordo com as restrições do problema, a combinação dos cromossomos dentro de um indivíduo determinam um *fitness* associado a ele. Esse valor é contabilizado na população, estabelecendo assim a denominada função objetivo do algoritmo [2,9].

O processo iterativo inicia-se com a determinação, de forma aleatória ou direcionada, da população inicial. Em sequência, para a criação de uma nova geração, são escolhidos alguns indivíduos para realizar a operação de cruzamento. Neste, os operadores genéticos de seleção, mutação e *crossover* atuam sobre os cromossomos para alterar, aprimorar e corrigir eventuais situações relacionados aos novos indivíduos criados [2,9].

A seleção é basicamente o processo de escolha dos elementos mais propensos a mutação em um indivíduo. A escolha pode ser realizada de forma ponderada, associando com o *fitness* relativo a cada cromossomo com sua propensão a ser escolhido ou por outro método que se adeque a situação em questão. Já o processo de mutação consiste em alterar as características de um cromossomo visando aprimorar sua qualidade ou corrigir erros ocasionados por outros operadores. Por fim, o *crossover* executa a função de cruzamento, no qual dois integrantes da população atual são escolhidos para que seções de seus códigos genéticos sejam permutadas entre eles, gerando assim novos indivíduos [2,9].

4. METODOLOGIA

Levando em consideração os estudos realizados e os resultados obtidos em trabalhos correlatos, a proposta da aplicação do AG mostrou-se promissora. Devido à complexidade da situação, o método do AG clássico não pode ser utilizado, sendo necessário a elaboração de um método adaptado do mesmo.

Em primeira instância, a função objetivo visa minimizar o *fitness* para encontrar a melhor solução, diferentemente da ideia proposta por Holland (1975), no qual o objetivo é maximizar a função objetivo [2,3,9]. Depois, algumas alterações nos operadores genéticos

foram implementadas para acelerar a convergência da solução, diminuindo o número de gerações necessárias para alcançar a meta estabelecida.

Ainda, a adaptação do AG foi planejada para atender as restrições que o caso da formulação da grade horário da UFPR apresenta.

Analisando o problema atual, formularam-se cinco situações que afetam o custo de cada designação ou limitam o método de busca, sendo elas:

1. O número de alunos em uma turma excede a capacidade da sala. Este caso tem a maior relevância dentro do problema, pois é a única situação que deve ser obrigatoriamente evitada.
2. O número de alunos em uma turma é muito menor que a capacidade da sala. Apesar de aparentemente não parecer um problema em questão, esse evento tem uma relação importante com a maneira que o método foi desenvolvido. O fator foi levado em consideração, pois ao alocar uma turma pequena a uma sala com capacidade elevada, há uma perda considerável de oportunidade para alocação das turmas com um grande número de alunos.
3. Uma turma com preferência por um bloco específico deve ser alocada, se possível, em uma sala deste bloco. Essa situação é específica da UFPR, no qual alguns cursos possuem blocos próprios, sendo assim, a situação deve ser considerada para o problema.
4. Uma turma com aula em diferentes períodos da semana deve ser, preferencialmente, alocada na mesma sala. Essa restrição é a menos relevante para o problema, ela apenas tem a função de facilitar a organização da grade de horários dos alunos e dos professores.
5. Uma turma de desenho ou laboratório não deve ser alocada em salas convencionais, assim como turmas convencionais não devem ser alocadas em salas de desenho ou laboratório. Esta é a única restrição que atua como limitante na alocação de dados, ou seja, diferente das descritas anteriormente, essa não afeta o fitness do indivíduo, mas sim no método de procura e alocação.

De acordo com as restrições, as funções de busca dentro do método, os operadores genéticos utilizados, cada nova geração apresenta, obrigatoriamente, um resultado melhor ou igual a sua população de origem. Sendo assim, diferente dos AG convencionais, no qual existem variações tanto positivas como negativas na geração de uma nova população, o método criado converge em sua totalidade de forma positiva, sem que haja acíves no gráfico da função objetivo pelas gerações.

4.1. DEFINIÇÃO DA POPULAÇÃO INICIAL

O processo de execução da metodologia produzida inicializa com a determinação da população inicial. Nesta, são alocados n turmas em m salas de forma aleatória, seguindo apenas a restrição de salas especiais como laboratórios ou ateliês. De acordo com a ordem em que os dados foram inseridos, através de um método simples de busca, as turmas vão sendo alocadas uma a uma em uma sala que ainda não tenha sido ocupada. O método de busca procura apenas salas com horários compatíveis com os horários da turma, dessa forma, além de facilitar os processos subsequentes por não gerar erros de compatibilidade de horário, o algoritmo tem sua velocidade e execução potencializada.

Esse procedimento se repete até que os p indivíduos da população inicial sejam preenchidos. Após a finalização desta etapa, o algoritmo inicia sua parte iterativa.

4.2. PROCESSO ITERATIVO

O procedimento de geração de novas populações é um processo iterativo que faz analogia à teoria evolucionária descrita por Darwin. Nela, cada geração adquire os aspectos de adaptação ao ambiente inerentes aos indivíduos da população anterior, fazendo com os novos indivíduos aumentem sua propensão sobrevivência ao meio [2,9]. Baseando-se nessa

metodologia, foi elaborado um processo iterativo que visa encontrar uma solução viável de forma eficiente e eficaz.

Na criação de uma nova população, o método usa uma adaptação dos operadores genéticos clássicos de crossover e mutação. Ainda, um novo conceito foi inserido ao método original, denominado de “filho ideal”.

4.2.1. Filho Ideal

Antes de descrever o processo iterativo e os operadores genéticos, é necessário explicar um novo conceito utilizado neste trabalho, pois este influencia diretamente na convergência e qualidade do resultado.

Ao analisar a estrutura do AG, notou-se que, para suprir esta demanda, dependendo do tamanho do problema, um número muito grande de iterações ou gerações são necessárias para se alcançar o resultado almejado [3]. Um fato que influencia esse evento é o fator probabilístico deste método, no qual há uma chance de que em algum momento, uma população diminua a qualidade da função objetivo em relação a sua predecessora [9]. Visando contornar essa dificuldade, entre outros fatores, foi desenvolvida uma metodologia denominada nesse trabalho de “filho ideal”. O fundamento desse elemento consiste em criar uma base de código genético para as futuras interações de crossover e mutação. Para tanto, o filho ideal incrementa seu código genético com os melhores cromossomos de cada novo indivíduo gerado e o próximo a ser criado, antes de passar pelos operadores genéticos, recebe todo o código genético do filho ideal.

Nesse contexto, o filho ideal é um indivíduo incompleto em relação aos seus cromossomos, ou seja, ele não possui uma estrutura genética completa. Contudo, como descrito anteriormente, a cada geração, seu código genético é incrementado, seguindo um critério de entrada delimitado por dois fatores criados para essa instância, a taxa de evolução e o limite de inserção.

A taxa de evolução indica quanto da estrutura genética do filho ideal está completa. Ainda, esta tem a função de aumentar o valor do limite de inserção, caso não haja mais acréscimo na taxa de evolução por um determinado período. Já o limite de inserção delimita a entrada de cromossomos no filho ideal seguindo um critério simples, se o cromossomo c possuir *fitness* menor que o limite de inserção, ele é implantado à estrutura do filho ideal. Dessa forma, com o auxílio dos operadores genéticos adaptados para esse método, é possível garantir que a geração subsequente apresente uma qualidade melhor que sua população criadora.

A tabela abaixo apresenta a variação da taxa de evolução (TE) e o limite de inserção (LI) para um exemplo com 10 indivíduos gerados em sequência.

Indivíduo	1	2	3	4	5	6	7	8	9	10
TE (%)	0	3	5	5	5	8	12	12	12	16
LI	1	1	1	1	2	2	2	2	3	3

Tabela 1: Exemplo da variação da taxa de evolução e limite de inserção. Fonte: O Autor.

Nota-se que, neste exemplo, após o 3º indivíduo a TE não apresenta incremento em seu valor por mais duas iterações, isso decorre pois, possivelmente, não há mais designações que possam gerar um *fitness* menor que o LI. Dessa maneira, o LI é forçado a ser acrescido em uma unidade, possibilitando assim novas inserções no código genético do filho ideal. O número de iterações necessárias para o incremento do LI dependerá das condições do problema e deverá ser estabelecido ao início do algoritmo, ou seja, é parametrizável. Este processo é repetido até que o último indivíduo seja criado. Uma solução muito boa seria o caso em que o filho ideal, ao final do algoritmo, alcançasse uma estrutura genética completa. Entretanto, esse fato não garante que o ótimo global do problema seja alcançado.

4.2.2. Crossover

Esse operador genético caracteriza-se como um dos fundamentos mais importantes para o AG, pois ele permite alcançar as possíveis singularidades no processo de designação. No método clássico, o crossover realiza a permutação de partes do código genético de dois indivíduos de uma população para gerar dois novos elementos, que pertencerão a uma nova geração. Esse processo, em sua maioria, demanda correções através do operador de mutação, sendo que, para o problema da alocação de turmas na UFPR, demandaria muito esforço computacional. Nesse sentido, uma alteração nesse operador genético foi implantada para evitar esse procedimento de correção póstuma.

Primeiramente, diferentemente do AG clássico, em vez do método gerar dois novos indivíduos, o procedimento criado gera apenas um. Esse novo componente da população, ao início da iteração, não possui código genético, sendo inicializado através da estruturação pelo filho ideal citado, após isso, aplica-se o crossover e mutação para obter sua estrutura completa.

Após o processo de alocação do código genético do filho ideal ao mesmo, o novo indivíduo sofrerá a ação do operador crossover. Para a execução deste, são escolhidos ao acaso dois elementos da população atual para realizar a troca de cromossomos. Nesse procedimento, ao invés de ser efetuado a permutação das estruturas genéticas sem critérios de entrada, o método adaptado apenas aloca os cromossomos que ainda não foram estabelecidos. De forma exemplificada, caso uma sala já tenha sido alocada a uma turma através da estruturação genética pelo filho ideal, mesmo que ela pertença ao segmento escolhido para o crossover, ela não sofrerá alteração em sua designação. Ainda, há outro critério de entrada que analisa a qualidade dos cromossomos dos indivíduos selecionados para o crossover. Como no método desenvolvido há apenas a criação de um indivíduo, é necessário determinar qual cromossomo será inserido entre as duas possibilidades dos elementos escolhidos no crossover. Dessa forma, se for possível a inserção, será escolhido o cromossomo que obtiver o menor custo de fitness, possibilitando a aplicação dos melhores elementos dentro da seção escolhida para o operador genético.

O esquema abaixo ilustra o funcionamento da operação de crossover descrito.

Seção genética do Indivíduo 1

1,34	23,4	4,01	2,61	32,1	10,9	8,03	2,05
------	------	------	------	------	------	------	------

Seção genética do Indivíduo 2

43,3	12,8	1,00	4,35	13,4	3,09	14,7	7,25
------	------	------	------	------	------	------	------

Seção correspondente do novo indivíduo

1,34	1,00	1,00	2,61	13,4	3,09	8,03	1,00
------	------	------	------	------	------	------	------

Ilustração 1: Operador crossover; Fonte: O Autor

No esquema acima, cada retângulo representa um cromossomo do indivíduo e cada linha representa uma seção do código genético dos indivíduos envolvidos no processo de crossover. Ainda, os valores nos cromossomos representam os respectivos custos do fitness relacionado a determinada designação.

Nota-se que o cromossomo escolhido é sempre aquele com o menor valor de fitness e caso o cromossomo já não tenha sido designado (célula com sombreamento amarelo) através do processo de filho ideal. Dessa forma, a criação do novo indivíduo é direcionada em relação ao seu custo de fitness.

Por fim, para proporcionar maior variabilidade e permitir ao algoritmo encontrar as singularidades na execução, a operação de crossover é realizada apenas em uma parte dos indivíduos selecionados, deixando para o último processo de estruturação, a mutação, a função de completar a estrutura genética faltante do novo indivíduo.

4.2.3. Mutação

Para o algoritmo desenvolvido, a mutação caracteriza-se como um intenso processo de busca, que visa encontrar uma alocação viável para os cromossomos sem designação, através de um processo de cinco estágios. Em cada estágio, há um critério de busca que analisa diversos critérios relacionados as restrições, sendo que se um cromossomo passar por todos os estágios sem resultado, não existirá alocação de um recurso a ele no indivíduo analisado.

A mutação tem um resultado excelente em relação a alocação, contudo, demanda muito tempo de execução. Por esse motivo, ela é apenas executada na parcela restante da estrutura do indivíduo que não for preenchida.

Em relação aos estágios de busca, estes são compilados quando o critério estabelecido na etapa predecessora não é atendido para todas as possibilidades de designação, sendo que, quando o critério é satisfeito, o procedimento retorna ao primeiro estágio, mas focado em outro cromossomo. Os critérios, em ordem de execução, são:

1. O destino da designação deve estar livre e ser o mesmo que a primeira alocação do tipo de recurso. Este critério, no problema da formulação da grade horária, visa resolver o problema em que uma turma com aulas em vários períodos da semana seja alocada em salas diferentes. Mesmo que esta restrição seja a menos relevante para o problema, ela deve ocupar o primeiro critério de busca, pois senão ela não seria verificada. O critério não é satisfeito quando a sala destino já está ocupada por outra turma.
2. O destino da designação deve estar livre e respeitar os critérios de preferência do recurso, assim o número de elementos do recurso não deve ultrapassar a sua capacidade, sendo que a diferença entre os valores não deve ultrapassar 15% da capacidade do destino.
3. O destino da designação deve estar livre e o número de elementos do recurso não deve ultrapassar a sua capacidade, sendo que a diferença entre os valores não deve ultrapassar 35% da capacidade do destino.
4. O destino da designação deve estar livre e respeitar os critérios de preferência do recurso, assim como o número de elementos do recurso não deve ultrapassar a capacidade desse.
5. O destino da designação deve estar livre.

Seguindo essa metodologia de busca, o indivíduo tem sua estrutura complementada. Após o fim desse processo, o algoritmo analisa todos os cromossomos do novo indivíduo e os insere ao filho ideal, caso o critério do limite de inserção seja atendido.

Por fim, para renovação da população, é realizada uma verificação do indivíduo de menor qualidade genética dentro da geração atual, sendo que este é removido para permitir a entrada do novo indivíduo a população.

4.2.4. Algoritmo do Processo Iterativo

Para $i = 1$ até G

*Para $j = 1$ até $TE * n_{Cni}$*

$C_{ni}(i, j) = C_{fi}(j)$

Próximo j

Para $k = a$ até b

Se $C_{ni}(i, k)$ não tiver alocação faça

Se $fitness[C_{co1}(i, k)] \geq fitness[C_{co2}(i, k)]$ faça

$C_{ni}(i, k) = C_{cc2}(i, k)$

Caso contrário faça

$C_{ni}(i, k) = C_{cc1}(i, k)$

Próximo k

Para $w = b$ até $nCni$
Verificação do Critério1
Se Critério1 foi estabelecido faça
 $Cni(i, w) = C1(i, w)$
Se não
Verificação do Critério2
Se Critério2 foi estabelecido faça
 $Cni(i, w) = C2(i, w)$
Se não
Verificação do Critério3
Se Critério3 foi estabelecido faça
 $Cni(i, w) = C3(i, w)$
Se não
Verificação do Critério4
Se Critério4 foi estabelecido faça
 $Cni(i, w) = C4(i, w)$
Se não
Verificação do Critério5
Se Critério1 foi estabelecido faça
 $Cni(i, w) = C5(i, w)$
Se não
Não é possível designar o $Cni(i, w)$
Próximo w
Para $z = 1$ até $nCni$
Se $fitness[Cni(i, z)] \leq LI$ faça
 $Cfi(z) = Cni(i, z)$

Próximo z

Próximo i

Onde: $i, j, k, w, z, a, b \in N$;

$Cni(i, x)$ representa a designação dos cromossomos do novo indivíduo;

$Cfi(x)$ representa a designação dos cromossomos do filho ideal;

$Cco1(i, x)$ e $Cco2(i, x)$ representam da designação dos cromossomos dos indivíduos escolhidos para o crossover;

$fitness[C]$ representa o valor do custo do fitness para a designação do cromossomo C ;

G representa o número gerações do algoritmo genético;

$nCni$ representa o número total de cromossomos em um indivíduo;

LI representa o limite de inserção para o filho ideal;

TE representa o taxa de evolução relacionado ao filho ideal;

Critério1 ao Critério5 representam os critérios dos estágios da mutação;

$C1(i, x)$ a $C5(i, x)$ representam a designação encontrada na busca realizada no processo de mutação;

5. IMPLEMENTAÇÃO E RESULTADOS

A implementação do AG adaptado foi realizada através do software Microsoft Visual Express 2010, em linguagem VB.net. Para a obtenção e alocação de dados, foi utilizada a plataforma Microsoft Office Excel.

Devido à grande quantidade de informações pertinentes a cada cromossomo, não foi possível utilizar o sistema binário para a alocação das turmas. Em contra partida, foram criadas duas planilhas que detinham todos os dados relevantes, em relação às turmas e as salas do problema e estas serviram como estrutura genética para a formação dos cromossomos.

Um grande desafio encontrado nesse trabalho foi a elaboração de um sistema que calculasse o *fitness* referente a cada turma, de forma que todos os problemas citados na seção 2 fossem abrangidos. Para tanto, a seguinte equação foi desenvolvida:

$$Fitness = 10^{(T-C)} + (C/30)^{\left[\frac{C-T}{15}\right]} + \left|\frac{Bf - Bs}{5}\right| + Ph$$

Onde: T = Número de alunos da turma; C = Capacidade da sala; Bf = posição do bloco preferencial referente a turma; Bs = posição do bloco em que a turma foi alocada; Ph = incremento relacionado ao problema de turmas com alocação em salas distintas em diferentes horários da semana.

Cada parte da função tem relação com um problema apresentado nesse trabalho, sendo que estes afetam em diferentes proporções o valor do *fitness* de acordo com a gravidade do evento em questão. Primeiramente, o fator " $10^{(T-C)}$ " faz analogia ao acontecimento em que uma turma possui mais alunos do que a capacidade da sala. Como este é um problema muito grave, o impacto na função é muito alto caso $T > C$. Na sequência tem-se " $(C/30)^{\left[\frac{C-T}{15}\right]}$ ", que pondera o acontecimento de salas muito grandes alocadas a turmas pequenas. Além do expoente que dita o impacto da diferença entre T e C, tem-se o fator C na base da exponencial em razão com um valor fixo. A escolha desse fator variável nessa posição se dá pelo fato de que salas maiores podem alocar mais turmas do que salas menores, ou seja, os tamanhos das salas influenciam de formas distintas a função, mesmo que tenham a mesma porcentagem de ocupação. O próximo elemento, $\left|\frac{Bf - Bs}{5}\right|$, faz referência ao problema de uma turma não estar alocada em seu bloco preferencial. O fator incrementa o *fitness* de acordo com a distância relativa entre o bloco preferencial e o bloco em que a turma foi situada naquele momento. Ainda, caso a turma não tenha bloco preferencial, o valor do elemento recebe uma quantidade fixa, para evitar que estes ocupem, inicialmente, as salas destinadas a turmas com preferência. Por fim, a variável Ph é um acréscimo pequeno ao valor do peso, caso ocorra o problema em uma turma seja alocada em diversas salas, em diferentes horários da semana.

O problema apresentou 2924 dados de entrada relacionados as turmas e seus respectivos horários e 5800 dados de entrada relacionados as salas e os períodos que estas estão disponíveis. A inserção de dados ao software desenvolvido era realizada através da importação de uma planilha com todas as informações pertinentes para o problema, como número de alunos, capacidade da sala e blocos preferenciais, por exemplo. Ainda, foi inserido ao programa um esquema da distribuição dos blocos para a que a função que calcula o *fitness* pudesse ponderar a distância entre os blocos preferencial e o bloco em que a turma foi situada.

Da forma que o algoritmo foi codificado, os resultados apresentam-se de duas maneiras: uma representando o *fitness* total e outra representando o *fitness* relativo da última população. O produto final é o indivíduo com o *fitness* mais baixo da última população. A solução ideal, embora impossível, teria um *fitness* relativo igual a 1, sendo assim, quanto mais próximo deste valor, melhor a qualidade da solução.

Diversos testes computacionais foram realizados, sendo que em cada teste um parâmetro era alterado, visando verificar seu impacto no resultado. Dessa maneira, alguns resultados obtidos são apresentados na tabela a seguir.

Nº de Gerações	Nº de indivíduos criados por geração	Nº de indivíduos em uma população	Taxa De Crossover ¹	Períodos sem evolução em TE ²	TE final	Problemas de Bloco ³	Problemas de Tamanho ⁴	Fitness Total	Fitness Relativo	Tempo de Execução Aproximado
150	3	25	150	6	94%	1039	15	80689	27,53	12 min
200	3	25	150	6	94%	986	17	77546	26,52	14 min
150	6	25	150	6	93%	1181	15	83118	28,42	13 min
200	3	100	200	6	95%	876	14	70908	24,25	12 min
200	3	100	150	9	96%	783	12	64080	21,91	13 min

Tabela 2: Resultados do AG desenvolvido; Fonte: O Autor

Taxa De Crossover¹ delimita o tamanho da seção de estrutura genética que será usada no operador crossover, o valor é expresso em número de turmas as serem alocadas;

Períodos sem evolução em TE² indica o número de indivíduos criados em que a taxa de evolução (TE) não sofre alteração, caso o algoritmo passe do valor expresso, o limite de inserção sofre o acréscimo de uma unidade;

Problemas de Bloco³ indica quantas turmas com preferência por blocos estão alocadas em salas de outros blocos;

Problemas de Tamanho⁴ indica quantas turmas estão alocadas em salas com menos capacidade que o número de alunos da turma ou turmas que não sem designação;

Pelos valores mostrados, nota-se que o número de gerações ou indivíduos criados não influencia positivamente o resultado caso o seu valor seja maior. Entretanto, variações nos períodos sem evolução da TE e o número de indivíduos em uma população apresentou resultados mais promissores. Esse fato decorre por dois motivos. O primeiro é a relação de variabilidade do problema, em função do número de indivíduos em uma população, sendo que quanto mais elementos existem em uma população, maior é a chance do processo de crossover encontrar singularidades em relação aos cromossomos.

O outro motivo está relacionado ao conceito do filho ideal aplicado ao algoritmo. Dependendo do período estabelecido para o incremento, maior é a possibilidade de cromossomos com fitness melhores serem inseridos ao filho ideal. Contudo, caso o período seja muito grande, além de demandar mais gerações para alcançar um resultado viável, não há garantia de que o resultado seja melhor. Isso acontece devido a um evento notado no desenvolvimento da metodologia do filho ideal, no qual nem sempre escolher os melhores cromossomos, inicialmente resultará em um melhor indivíduo. Ainda, nota-se que em todos os resultados finais, há pelo menos 12 problemas de tamanho. Essa situação está relacionada a um problema de capacidade do campus Centro Politécnico da UFPR, em que em alguns horários, não há capacidade suficiente para a demanda de salas.

Em relação à comparação com o método atual aplicado para a formulação da grade horária na UFPR, o AG obteve resultados mais relevantes em todas as suas simulações. A tabela abaixo apresenta o resultado do método atual e o melhor resultado encontrado com o AG aplicado.

	Problemas de Bloco	Problemas de Tamanho	Fitness Total	Fitness Relativo
Algoritmo Genético	783	12	64080	21,91
Redes Neurais Artificiais	1348	30	117237	40,10

Tabela 3: Comparação dos resultados entre o método atual e o desenvolvido; Fonte: O Autor.

Pela Tabela 3, percebe-se resultados melhores do método desenvolvido em todos os fatores, sendo que o AG aplicado obteve quase o dobro de eficácia, em função da metodologia atual. Em relação a comparação entre os tempos de execução dos processos, devido à falta de informações sobre o RNA, não foi possível estabelecer um parâmetro comparação. Contudo, o sistema realizado possui um tempo de execução, em média, de 14 min. Desta forma, levando em consideração que a grade horária terá duração de 6 meses, o método apresentou resultados mais que satisfatórios para o estudo realizado.

6. CONCLUSÃO

Neste artigo, foi desenvolvido uma adaptação do AG para resolver o problema da grade horária do campus Centro Politécnico da UFPR. O algoritmo foi implementado com foco na velocidade de execução e eficácia do método. Para tanto, vários testes foram realizados visando aprimorar seu desempenho e corrigir os eventuais ajustes no modelo. Ainda, para validação do resultado, este foi comparado com os resultados obtidos com o método atual usado na formulação da grade horário, obtendo grande melhoria em relação a eficácia em função dos parâmetros analisados. Desta maneira, foi demonstrado a qualidade da

aplicação do AG para problemas de designação de turma para salas, onde há variabilidade e complexidade das restrições.

O objetivo de criar um método aplicável e eficaz através de uma adaptação do AG foi alcançado, sendo que o método pode ser utilizado em qualquer situação de designação por períodos que se assemelhe ao problema analisado.

7. REFERÊNCIAS BIBLIOGRÁFICAS

- [1] ABRAMSON, D. Constructing School timetables using simulated annealing: sequential and parallel algorithms. *Management Science*, vol.37, pp. 98– 113. 1991.
- [2] BATTERSBY, A.; EL-MIHOUB, T. A.; HOPGOOD, A. A.; NOLLE, L. Hybrid Genetic Algorithms: A Review. 2006. http://www.engineeringletters.com/issues_v13/issue_2/EL_13_2_11.pdf. Acessado em 15/04/2014 21:59.
- [3] BELIGIANNIS, G. N.; MOSCHOPOULOS, C. N.; KAPERONIS G. P.; LIKOTHANASSIS S. D. Applying evolutionary computation to the school timetabling problem: The Greek case. *Computers & Operations Research*, vol. 35, no. 4, pg. 1265-1280, 2008.
- [4] BLAS, L. E. A.; SANZ, S. S.; GARCÍA, E. G. O.; FIGUERAS, A. P.; BELLIDO, Á. M. P. A hybrid grouping genetic algorithm for assigning students to preferred laboratory groups. *Expert Systems with Applications*, vol. 36, no. 3, pg. 7234 -7241, 2009.
- [5] COLORNI, A; IRIDA, M; MANIEZZO V. Metaheuristics for High School Timetabling. *Computational Optimization and Applications*, vol. 9, no. 3, pg.275-298, 1998.
- [6] COSTA, D. A tabu search algorithm for computing an operational timetable. *European Journal of Operational Research*, vol. 76, no. 1, pg. 98– 110. 1994.
- [7] DUAN, Y.; ZHONG, Y.; LI, Y. Application Research on FSDM-based GA in Optimizing Curriculum Schedule Model in Universities. *Information and Engineering*, vol.4, no.2, pg.217-228, 2012.
- [8] GOLDBERG, D. E. Genetic algorithms in search, optimization and machine learning. Massachusetts: Addison-Wesley; 1989. 412 páginas.
- [9] MELANIE, M. An Introduction to Genetic Algorithms. MA: A Bradford Book The MIT Press; 1999, 158 páginas. <https://svn-d1.mpi-inf.mpg.de/AG1/MultiCoreLab/papers/ebook-fuzzy-mitchell-99.pdf>. Acessado em 15/04/2014 21:59.
- [10] MOZOS, R. S.; SANZ, S. S.; CUMPLIDO, M. D.; CALZON, C. B. A two-phase heuristic evolutionary algorithm for personalizing course timetables: a case study in a Spanish university. *Computers and Operations Research*, vol. 32, no. 7, pg.1761-1776. 2005.
- [11] SCHAERF, A. A survey of automated. *Artificial Intelligence Review*, vol.13, no.2, pg.87-127. 1999.
- [12] STEINER, M. T. A. Meta-Heurísticas em Pesquisa Operacional. 1 ed. Curitiba - PR: Omnipax Editora Ltda, v. 1, p. 21-31. 2013.
- [13] VALOUXIS, C.; HOUSOS, E. Constraint programming approach for school timetabling. *Computers & Operations Research*, vol.30, no.10, pg. 1555–1572. 2003.