



Universidade Federal de Pernambuco

Centro de Informática

Graduação em Ciências da Computação

SAPA - Sistema de Alocação de Professores e Agendamento de Aulas

Thales Vinicius Gomes Pereira

Trabalho de Graduação

Recife

23 de dezembro de 2011

Universidade Federal de Pernambuco

Centro de Informática

SAPA - Sistema de Alocação de Professores e Agendamento de Aulas

Thales Vinicius Gomes Pereira

tvgp@cin.ufpe.br

Trabalho de Graduação apresentado no Centro de Informática (CIn) da Universidade Federal de Pernambuco (UFPE) como requisito parcial para obtenção do grau de Bacharel em Ciências da Computação.

Orientador: Aluízio Fausto Ribeiro Araújo

Recife

23 de dezembro de 2011

Resumo

A geração de tabelas de horários consiste em montar uma tabela de horários onde, em cada lacuna de tempo dia x hora eventos são agendados para acontecer em determinados locais. Esta geração consiste um dos maiores problemas enfrentados por instituições de ensino para se montar a planilha de aulas de um período letivo bem como organizar em quais dias ocorrerão as provas de cada matéria. Neste trabalho abordamos como modelar o problema e apresentamos a técnica de algoritmos genéticos com busca guiada para a geração automatizada de horários, mostrando ao final os resultados obtidos.

Abstract

Timetabling consists in making a timetable where, at each day x hour timeslot, events are scheduled to happen in a given place. This timetable making is one of the biggest problems faced by teaching institutions in order to build a complete schedule of the classes and tests of each subject for the whole year. In this work we deal with the problem modelling and present the guided search genetic algorithm technique to make an automatic timetable maker, showing the achieved results at the end.

Agradecimentos

Pela conclusão deste trabalho e deste curso, agradeço primeiramente a Cristo Jesus, por ter sempre me amparado e me sustentado com sua mão em meio aos vários momentos difíceis que passei na minha vida ao longo do curso.

Agradeço a todos os professores que me ensinaram ao longo dessa jornada e, em especial, ao meu orientador, o professor doutor Aluizio Fausto Ribeiro Araújo, por ter depositado em mim sua confiança e acreditado na minha capacidade, e por estar sempre disposto a me ouvir e me orientar com relação ao projeto aqui desenvolvido.

Agradeço aos meus familiares, a meus pais Lúcio José Gomes Pereira e Vera Lúcia Paula da Silva Gomes, por terem me apoiado em especial neste final de curso.

Por último, mas não menos importante, agradeço à minha noiva Paula de Santana Bezerra, por ter me dado apoio e se empenhado em me ajudar em meio aos diversos problemas que passei na minha vida pessoal. Sem ela eu não teria conseguido chegar até aqui.

Sumário

1. Capítulo I – Introdução.....	8
1.1. Objetivos	8
1.2. Estrutura do Documento.....	9
2. Capítulo II – Definição do Problema.....	10
2.1. Restrições.....	10
3. Capítulo III – Técnicas Utilizada para Resolução do Problema	13
3.1. Coloração de Grafo	13
3.2. Busca Local Iterativa	13
3.3. Têmpera Simulada	14
3.4. Colônia de Formigas.....	15
3.5. Busca Tabu	16
3.6. Algoritmos Genéticos.....	16
3.7. Técnica Escolhida	17
4. Capítulo IV – Modelagem.....	18
4.1. O Modelo de Olivia Rossi-Doria.....	18
4.2. Modelagem Proposta.....	20
4.3. Requisitos de Entrada e Saída	21
4.3.1. Entradas	21
4.3.1.1. Professor	21
4.3.1.2. Disciplina	21
4.3.1.3. Aula Variável	23
4.3.1.4. Aula Fixa	23
4.3.1.5. Sala de Aula	23
4.3.2. Saídas	24
4.3.2.1. Agendamento por Período	24
4.3.2.2. Agendamento por Sala	24
4.3.2.3. Agendamento por Professor	24
5. Capítulo V – Algoritmo Genético de Busca Guiada	25
5.1. Estrutura MEM.....	25
5.2. Geração de Filhos.....	27
5.2.1. Geração de Filhos Utilizando-se a MEM	27
5.2.2. Geração de Filhos por Cruzamento	28

5.3.	Mutação	28
5.4.	Busca Local	28
6.	Capítulo VI – Resultados Obtidos	32
6.1.	Horário Atual	32
6.1.1.	Agendamento	32
6.2.	Resultados Obtidos	44
6.2.1.	Penalidades	44
6.2.2.	Considerando Disciplinas Externas com Horários Fixos.....	44
6.2.2.1.	Parâmetros utilizados	44
6.2.2.2.	Restrições Violadas.....	44
6.2.2.3.	Agendamento	46
6.2.3.	Considerando Todas as Disciplinas sendo Lecionadas no CIn	57
6.2.3.1.	Parâmetros utilizados	57
6.2.3.2.	Restrições Violadas.....	57
6.2.3.3.	Agendamento	58
7.	Capítulo VII – Considerações Finais e Trabalhos Futuros	69
7.1.	Trabalhos Futuros	70
	Referências Bibliográficas	73

Índice de Tabelas

Tabela 1 - Restrições Obrigatórias	11
Tabela 2 - Restrições Opcionais	12
Tabela 3 - Penalidade das restrições violadas.....	44

1. Capítulo I – Introdução

Toda instituição precisa que seus profissionais trabalhem em conjunto para melhor atender seu público alvo. Por outro lado, os trabalhadores da instituição também necessitam de folgas, intervalo para almoço e férias. Como ser humano, nenhum profissional pode ficar à disposição da instituição e do público 24/7. Desta forma, toda instituição precisa alocar as horas de trabalho de seus profissionais de forma a melhor servir a empresa e o público.

Dada a descrição acima, podemos definir o Problema do Agendamento como sendo o problema de se alocar os profissionais e os recursos de uma instituição de modo a atender simultaneamente e da melhor forma possível os interesses da instituição, do público alvo, e dos funcionários da instituição.

Neste trabalho de graduação especificamente, iremos nos focar no problema de agendamento de aulas em instituições de ensino. Nosso objetivo é desenvolver um método automatizado por computador para a geração de um quadro de aulas que seja considerado bom tanto por professores e alunos como também pela própria instituição.

A geração de tabelas de horários é considerada um problema difícil de otimização combinatória enfrentado por universidades por todo o mundo nos dias atuais. Dado o número crescente de estudantes em universidades, um grande número de cadeiras é oferecido a cada período letivo. Cada disciplina oferecida tem uma determinada quantidade de vagas, e cada sala da instituição tem uma capacidade de acomodação de alunos e recursos próprios da sala para apoio didático. Além disso, não é bom alocar uma disciplina com poucas vagas em um local muito grande, pois implicaria em utilização ineficiente da sala. As disciplinas oferecidas a alunos de um semestre não podem ter choques de horários entre si, nem os professores, podem ensinar mais de uma cadeira ao mesmo tempo. Estes itens tornam não trivial o processo de geração de tabela de horários. De fato, tal processo via de regra é feito por uma ou mais pessoas e costuma levar muito tempo, podendo demandar dos responsáveis várias horas de trabalho ao longo de algumas semanas em instituições de grande porte até que se chegue a uma alocação de qualidade satisfatória.

1.1. Objetivos

Neste trabalho de graduação, temos por objetivo desenvolver um algoritmo que seja capaz de gerar automaticamente uma tabela de horários que atenda às exigências reais do curso de Engenharia da Computação do Centro de Informática da Universidade Federal de Pernambuco. Vários métodos já foram propostos na literatura, como técnicas de otimização baseada em colônia de formigas, coloração de grafo, têmpera simulada, e algoritmos genéticos.

O modelo por nós desenvolvido utilizou a técnica de algoritmos genéticos hibridizada com uma heurística de busca local, também conhecida como algoritmos genéticos de busca guiada. Vários testes foram realizados considerando cenários diferentes e diversas possibilidades de ajustes dos parâmetros. Por fim, conseguimos ajustar os parâmetros de forma a obter resultados melhores dos que os obtidos por especialistas humanos e com um tempo de execução muito baixo, mostrando assim a viabilidade da solução proposta em ser implantada no Centro de Informática para gerar os horários do curso de Engenharia da Computação.

1.2. Estrutura do Documento

No segundo capítulo veremos uma definição mais formal do problema e serão apresentadas e classificadas as restrições do problema. O terceiro capítulo dá uma visão geral das técnicas já empregadas para solucionar o problema e indicaremos a técnica escolhida. No quarto capítulo abordaremos conceitualmente a modelagem dos dados envolvidos no problema e quais as entradas e saídas providas pelo nosso sistema. É válido lembrar que essa modelagem independe da técnica utilizada para resolver o problema. A modelagem não trata da resolução do problema, mas sim de como representá-lo formalmente. No quinto capítulo discutiremos a fundo a abordagem utilizada para resolver o problema, explicando cada passo do algoritmo genético de busca guiada. No sexto capítulo mostraremos como está a situação atual dos horários do Centro de Informática da Universidade Federal de Pernambuco e quais as soluções propostas pelo sistema. No sétimo capítulo faremos observações sobre os testes e resultados obtidos e discutiremos sugestões de melhorias futuras do sistema.

2. Capítulo II – Definição do Problema

Nas instituições de ensino, especialmente universidades, o problema de agendamento possui duas vertentes principais: o Problema de Geração de Tabelas de Horários para Cursos Universitários (UCTP – University Course Timetabling Problem) e o Problema de Geração de Tabelas de Horários de Provas (ETP – Exam Timetabling Problem) – Rossi-Doria et al (2003). O UCTP consiste em alocar recursos a eventos em determinados locais e horários da tabela que deverão ser repetidos semanalmente ao longo do período letivo. Os recursos englobam recursos de pessoal, como professores e alunos, e de material, como um equipamento que deve ser levado ao evento. Os eventos englobam aulas teóricas, aulas práticas, seminários, entre outros. Os locais podem ser salas de aula, laboratórios, ou um local específico, como uma sala cirúrgica para o curso de Medicina ou uma quadra poliesportiva para o curso de Educação Física. Os horários da tabela representam os intervalos de tempo nos quais os eventos irão ocorrer. O ETP consiste em montar uma tabela de provas para os alunos, levando em consideração fatores como quantos dias serão necessários para agendar todas as provas de modo que todos os alunos possam fazer as provas, ou seja, nenhum aluno tem mais de uma prova agendada para o mesmo dia e horário, e evitando que os alunos tenham mais de uma prova por dia. No nosso caso, trataremos a geração da tabela de horário para as aulas – UCTP.

2.1. Restrições

Gerar uma tabela de horários de boa qualidade requer satisfazermos certas restrições. Uma tabela pode ser considerada ótima se todas as restrições exigidas forem satisfeitas. Segundo Rhydian Lewis restrições para este tipo de problema podem ser classificadas em obrigatórias e opcionais.

a) Restrições obrigatórias

Também conhecidas como restrições rígidas (*hard constraints*), são aquelas cuja não-satisfação implicará obrigatoriamente na existência de uma solução inviável. Por exemplo, podemos tomar o caso da restrição que diz que duas aulas de uma turma não podem ocorrer simultaneamente. Caso essa restrição não seja obedecida, o quadro de aulas gerado obrigará os alunos a estarem em duas aulas ao mesmo tempo, o que obviamente é impossível. As restrições obrigatórias que iremos considerar estão listadas na Tabela 1. Tais restrições são baseadas no trabalho de Rhydian Lewis, Rossi-Doria et al (2003), e Yang e Jat (2011).

b) Restrições opcionais

Também conhecidas como restrições flexíveis (*soft constraints*), são aquelas que devem preferencialmente ser satisfeitas, porém a sua não-satisfação não causará a inviabilidade de uma solução. Como exemplo podemos tomar o caso do turno das aulas. É preferível que as aulas de um mesmo período letivo sejam sempre agrupadas, pois isso diminui a quantidade de vezes que os alunos precisam ir e vir da instituição, como também os possibilita a ter um estágio ou emprego ou fazer um curso no turno livre. As restrições opcionais que iremos considerar estão listadas na Tabela 2. Tais restrições são baseadas no trabalho de Rhydian Lewis, Rossi-Doria et al (2003), e Yang e Jat (2011).

Dependendo das especificidades de cada caso, por exemplo, poucas salas disponíveis, nem sempre será possível satisfazer todas as restrições impostas ao sistema. Não obstante esta dificuldade, devemos satisfazer as restrições obrigatórias para não gerar um horário inviável. Assim, algumas restrições opcionais podem não ser satisfeitas para que tenhamos horários viáveis.

As restrições obrigatórias podem se resumir nas ideias de “ninguém pode estar em dois lugares ao mesmo tempo” e “uma sala não pode ter duas aulas ao mesmo tempo”. São restrições cujas violações jamais poderiam ser colocadas em prática, dada a própria contradição da física. Por outro lado, as restrições opcionais lidam com o conforto dos envolvidos. São restrições que envolvem, por exemplo, um aluno ter aula apenas de manhã ou de tarde. Um aluno poderia muito bem ter aulas espalhadas ao longo do dia, porém isso lhe causaria um maior cansaço cotidiano e um certo desconforto de ir à universidade pela manhã, ter apenas uma aula, e depois esperar até o fim da tarde pela outra aula, ou então ter uma dia com todo o turno da manhã e da tarde repleto de aulas.

Tabela 1 - Restrições Obrigatórias

ID	Restrição
ROb 1	Dadas duas disciplinas obrigatórias de um mesmo período elas não podem ser alocadas na mesma lacuna de tempo.
ROb 2	Nenhum professor pode ser alocado para dar mais de um evento na mesma lacuna de tempo.
ROb 3	Nenhuma sala pode ser alocada a mais de um evento na mesma lacuna de tempo.
ROb 4	Cada disciplina só pode ser alocada em uma sala que tenha capacidade para comportar a quantidade de alunos esperada pela disciplina.
ROb 5	Cada disciplina só pode ser alocada em um turno no qual ela está autorizada a ser realizada.
ROb 6	Caso a disciplina tenha mais de um evento por semana, os eventos não podem ocorrer no

	mesmo dia.
ROb 7	Cada professor contém uma lista indicando as lacunas de tempo onde ele está impossibilitado de participar de um evento. Essas lacunas devem ser evitadas.
ROb 8	Cada sala contém uma lista indicando em que dias e horários cada sala estará indisponível. Essas lacunas de tempo devem ser evitadas.

Tabela 2 - Restrições Opcionais

ID	Restrição
ROp 1	Disciplinas eletivas de um mesmo perfil não devem ser alocadas na mesma lacuna de tempo.
ROp 2	Cada professor contém uma lista indicando as lacunas de tempo nas quais ele não gostaria de dar aula. Essas lacunas devem ser evitadas.
ROp 3	Caso a disciplina possa ocorrer em mais de um turno e for informado um turno preferencial, ela deve ser alocada nesse turno.
ROp 4	Caso a disciplina tenha mais de um evento por semana, deve haver pelo menos 1 dia de folga entre os dias dos eventos.
ROp 5	Caso a disciplina possua mais de uma aula por semana, elas devem alternar os horários. Ex: se em um dia uma das aulas começa às 8:00hs e termina às 10:00hs, a outra aula deve começar às 10:00hs e terminar às 12:00h.

3. Capítulo III – Técnicas Utilizada para Resolução do Problema

Problemas de geração de tabelas de horários são geralmente problemas de otimização NP-completos de natureza combinatória e não há um algoritmo ótimo que gere soluções em tempo razoável (Rossi-Doria et al (2003)). Devido à complexidade do problema, a maior parte dos estudos se concentra em algoritmos heurísticos para tentar chegar a boas soluções por meio de aproximações.

As técnicas apresentadas a seguir constituem as principais vertentes utilizadas na resolução do UCTP (Rossi-Doria et al (2003)). Isso porém, não significa que elas não possam ser combinadas com outras técnicas ou mesmo entre si (Yang e Jat (2011)).

3.1. Coloração de Grafo

Na modelagem de coloração de grafo (Cheong et al. 2003) os dias da semana e horários disponíveis são agrupados em pares, por exemplo, segunda-feira no horário das 10:00h às 11:00h, terça-feira no horário das 10:00h às 11:00h, e quarta-feira no horário das 15:00h às 16:00h, e cada par é representado por uma cor diferente dos demais pares. Considera-se cada vértice do grafo uma aula, e cada aresta entre dois vértices indica que ambas as aulas não podem ocorrer simultaneamente, ou seja, no mesmo par dia da semana e horário. Após a montagem do grafo, o problema de agendamento resume-se a colorir todos os vértices de grafo de forma que nenhum par de vértices ligados por uma aresta tenha os dois vértices com a mesma cor. A solução gerada é então o grafo colorido, onde cada aula terá seu horário determinado pela associação do vértice que a apresenta com a cor ou horário com o qual ele foi colorido.

3.2. Busca Local Iterativa

Busca local iterativa (Iterated Local Search) se baseia na simples e poderosa ideia de melhorar um procedimento de busca local ao prover novas soluções iniciais obtidas através de perturbação de uma solução atual (Rossi-Doria et al). E geral, os novos resultados são melhores que aqueles obtido por inicialização aleatória. Para se aplicar busca local iterativa, quatro componentes precisam ser especificados:

a) Geração de solução inicial

Gera uma solução inicial aleatória s_0 usando uma distribuição uniforme. Nenhuma informação específica do problema é utilizada.

b) Perturbação

Modifica a solução atual levando a uma solução s' indeterminada. As perturbações são: mudar a lacuna de tempo de um evento, trocar a lacuna de tempo de 2 eventos entre si, e trocar a lacuna de tempo de 3 eventos entre si.

c) Busca local

Retorna uma solução melhorada de s' .

d) Critério de Aceitação

Decide em qual solução será aplicada a próxima perturbação. Os critérios de aceitação implementados por Rossi-Doria et al (2003). foram: “Busca Aleatória”, onde uma solução da busca local é sempre aceita, “Aceito se melhor”, onde a solução é aceita se for melhor que a atual, e “Têmpora Simulada” onde a solução sempre é aceita se for melhor e, caso seja pior, é aceita com uma probabilidade p .

3.3. Têmpora Simulada

Têmpora simulada (Simulated Annealing) é uma busca local inspirada no processo de recozimento definido na física. Esta técnica é amplamente utilizada para solucionar problemas de otimização combinatória, especialmente para evitar ficar preso em ótimos locais ao utilizar métodos de busca locais mais simples. Isto é feito da seguinte forma: um movimento que melhore a solução é sempre aceito enquanto um que a piora é aceito de acordo com uma probabilidade que depende do montante de deterioração do valor da função de avaliação, de modo que quanto pior for o movimento, menos provavelmente ele será aceito. Um movimento é aceito de acordo com uma distribuição probabilística, dependendo da temperatura virtual T , conhecida como a distribuição Metropolis. A têmpora simulada leva em consideração os seguintes fatores:

a) Temperatura

O parâmetro de temperatura T , que controla a probabilidade de aceitação, pode variar ao longo do processo de busca. Uma maior temperatura indica um relaxamento das restrições, dando uma maior probabilidade de aceitação de um movimento que piore a situação atual.

b) Temperatura inicial

Nos experimentos de Rossi-Doria et al (2003). duas possibilidades foram consideradas. A primeira é usar uma temperatura que provê uma probabilidade de $1/e$ de aceitação de um movimento que piore em até 2% o valor de avaliação de uma solução. A outra alternativa é pegar amostras de 100 vizinhos, tirar a média das variações das amostras e multiplicar esse valor por um dado fator para obter a temperatura inicial.

c) Duração da temperatura

O número de iterações nas quais a temperatura é mantida proporcional ao tamanho da vizinhança. Johnson et al. apontou esse fator como necessário para se atingir soluções de alta qualidade.

d) Resfriamento

Indica como e quão rápido a solução diminui o relaxamento das restrições e converge para uma solução final. Um resfriamento mais lento geralmente encontra soluções melhores, pois o sistema como um todo tende a se estabilizar mais facilmente. Aqui Rossi-Doria et al (2003). utilizaram também uma estratégia de reaquecimento do sistema para evitar ótimos locais.

e) Exploração da vizinhança

Duas estratégias foram consideradas por Rossi-Doria et al (2003). para gerar uma solução vizinha: A primeira estratégia considera os movimentos N_1 (mudar a lacuna de tempo de um evento) e N_2 (trocar as lacunas de tempo de 2 eventos) na mesma ordem que a busca local. Na estratégia apenas eventos envolvidos em violação de restrições são considerados. Mas diferentemente da busca local, após tentar todos os movimentos possíveis a um evento o algoritmo não para, mas continua com uma ordem diferente de eventos. A segunda estratégia descarta a busca local e utiliza uma estratégia de seleção de movimentos completamente aleatória. A cada passo um movimento proposto é gerado a partir da combinação de N_1 e N_2 .

3.4.Colônia de Formigas

Otimização de colônia de formigas é uma meta heurística proposta por Dorigo et al. A inspiração da colônia de formigas é o comportamento de formigas reais. O ingrediente básico da colônia de formigas é um mecanismo de construção de uma solução probabilística baseado na concentração de feromônio.

A cada iteração do algoritmo, cada uma das n formigas constrói, evento por evento, um agendamento completo. Para se fazer um agendamento de um evento em uma lacuna de tempo, uma formiga pega o próximo evento de uma lista pré-ordenada e escolhe probabilisticamente uma lacuna de tempo para ele guiada por 2 tipos de informação: informação heurística, que é uma avaliação das violações de restrição causadas por fazer o agendamento dados os agendamentos já feitos, e informação dos feromônio das iterações prévias do algoritmo.

A informação dos feromônio é representada por uma matriz de valores $t: E \times T \rightarrow R \geq 0$, onde E é o conjunto de eventos e T o conjunto de lacunas de tempo. Esses valores são inicializados com um parâmetro t_0 inicial e atualizados através de regras locais e globais. Geralmente, um par evento-lacuna de tempo que foi uma boa solução no passado possuirá maior valor de feromônio, e

consequentemente maior chance de ser escolhido novamente no futuro. Ao final da construção iterativa, a associação eventos-lacunas de tempo é convertida em uma solução candidata. Esta solução candidata é melhorada através de uma rotina de busca local. Depois de todas as m formigas terem gerado suas soluções candidatas é feita uma atualização global na matriz de feromônio utilizando-se a melhor solução encontrada desde o começo. Esse processo de construção se repete até que se atinja o limite de tempo.

3.5. Busca Tabu

Busca Tabu (*Tabu Search*) é uma meta-heurística que se baseia em estruturas de memória especializadas para evitar ficar presa em ótimos locais e alcançar um balanceamento efetivo entre intensificação e diversificação. A busca Tabu se mostrou muito poderosa em encontrar soluções de alta qualidade para problemas de otimização combinatória computacionalmente difíceis em uma variedade de aplicações. Mais precisamente, a busca Tabu permite a busca explorar soluções que não diminuam o valor da função objetivo, mas somente nos casos onde as essas soluções não são proibidas. Isto geralmente é obtido guardando-se as últimas soluções em termos de movimentos utilizados para transformar uma solução em outra. Quando um movimento é executado, o movimento reverso é considerado um tabu pelas próximas i iterações, onde i é o comprimento da lista de tabus. Uma solução é proibida se ela for obtida aplicando-se um movimento tabu na solução atual.

Proibimos um movimento se ao menos um dos eventos envolvidos foi movido em menos de i iterações passadas. O comprimento da lista de tabus é definido pelo número de eventos dividido por uma constante k . O objetivo desta lista é reduzir a probabilidade de se gerar ciclos e reduzir o tamanho da vizinhança para uma exploração mais rápida.

3.6. Algoritmos Genéticos

Algoritmos genéticos (AGs) propõem um modelo computacional do processo de evolução natural. AGs operam com uma população de possíveis soluções, chamadas de indivíduos, formados por cromossomos. Cada indivíduo é composto por uma série de variáveis do problema, chamadas de genes, e a cada indivíduo é atribuído um valor de aptidão, que indica quão boa a combinação dos valores das variáveis contidas nos seus genes é. Os AGs compreendem 3 estágios principais: seleção, reprodução e substituição. No estágio de seleção os indivíduos mais aptos possuem maior chance de reproduzir do que os menos aptos, da mesma forma que na seleção natural. Na reprodução os operadores de cruzamento e mutação são aplicados. No cruzamento (*crossover*), a solução filha gerada é uma combinação dos genes das soluções de seus pais. Este filho possui probabilidade de sofrer mutação. Na mutação, a solução gerada previamente sofre uma alteração aleatória em pelo

menos um dos seus genes. Finalmente, a população original de indivíduos é substituída pela nova população criada. Geralmente a população original não é totalmente descartada como acontece no mundo real. Em vez disso, são escolhidos os indivíduos mais aptos de ambas as populações de pais e de filhos e os demais, menos aptos, são descartados. A melhoria da qualidade das soluções é meta do estágio de seleção, enquanto a variação da população ocorre na fase de reprodução.

3.7. Técnica Escolhida

Rossi-Doria et al (2003). compararam diferentes meta-heurísticas na resolução do UCTP. Eles concluíram que AGs convencionais não apresentaram bons resultados. De fato, o resultado das pesquisas feitas por eles apontaram AGs como sendo o segundo pior de todos os modelos testados para se resolver o UCTP. Entretanto modelos de AGs melhorados, como o Algoritmo Genético de Busca Guiada (AGBG) proposto por Yang e Jat (2011) tem se mostrado promissores. Nos AGBGs uma estratégia de busca orientada é utilizada para gerar os descendentes da população baseados em uma estrutura de dados adicional. Tal estrutura será apresentada mais adiante. Os AGBGs mostraram resultados iniciais promissores nos experimentos realizados por estes autores. Segundo as comparações feitas por Yang e Jat (2011), o AG modificado proposto por eles em sua pior execução conseguiu ser consideravelmente melhor que os outros métodos comparados. O AG proposto por Yang e Jat (2011) é do tipo *Steady State*, onde apenas 1 indivíduo é criado a cada geração e o pior indivíduo da população é descartado. Isso garante que a convergência dos indivíduos não seja rápida demais, evitando que as soluções fiquem presas em um ótimo local. A solução proposta neste TG é baseada no AGBG abordagem que é adaptado para melhorar a eficácia e eficiência do modelo proposto.

4. Capítulo IV – Modelagem

4.1.0 Modelo de Olivia Rossi-Doria

O modelo proposto por Rossi-Doria et al (2003). abrange uma grande quantidade de elementos envolvidas no UCTP e pode ser utilizado com qualquer uma das técnicas descritas acima. Os autores empregaram todas as abordagens descritas, exceto a coloração de grafo, utilizando a modelagem de dados propostas por eles, para comparar as técnicas e analisar vantagens e desvantagens de cada uma. O modelo de Rossi-Doria é utilizado por outros autores ao se propor uma nova solução, tal como fizeram Yang e Jat (2011) (citar). Portanto, para entendermos melhor a modelagem do problema, partiremos do modelo de Rossi-Doria e colegas.

Segundo o modelo proposto por Rossi-Doria et al (2003). o UCTP consiste em: um conjunto de n eventos $A = \{a_1, \dots, a_n\}$ a serem agendados em 45 lacunas de tempo $T = \{t_1, \dots, t_{45}\}$. Estes números foram determinados levando-se em consideração uma semana útil de 5 dias onde cada dia possui 9 lacunas de tempo que podem ser preenchidas (8:00h-9:00h, ..., 17:00h-18:00h). Um conjunto de m salas disponíveis $S = \{s_1, \dots, s_m\}$ onde os eventos podem ocorrer, um conjunto de k estudantes $E = \{e_1, \dots, e_k\}$ que participam dos eventos, e um conjunto de l recursos disponíveis $R = \{r_1, \dots, r_l\}$ que são satisfeitos pelas salas e requisitados pelos eventos.

Além disso, o relacionamento entre tais conjuntos se dá através de 5 matrizes:

- a) A primeira matriz, $\mathbf{A}_{k,n}$, chamada de matriz estudantes-eventos, indica quais eventos são assistidos por quais alunos. Em $\mathbf{A}_{k,n}$ o valor de a_{ij} é 1 se o i -ésimo estudante participar do j -ésimo evento, ou 0 se o estudante não participar do evento.
- b) A segunda matriz, $\mathbf{B}_{n,n}$, chamada de matriz de conflito de eventos, indica quais eventos não podem ocorrer em conjunto. Em $\mathbf{B}_{n,n}$, valor de b_{ij} é 1 se o i -ésimo evento não puder ocorrer em paralelo com o j -ésimo evento, ou 0 se ambas puderem ocorrer simultaneamente. Essa matriz ajuda a identificar rapidamente se duas disciplinas podem ou não ser alocadas na mesma lacuna.
- c) A terceira matriz, $\mathbf{C}_{m,l}$, chamada de matriz salas-recursos, indica quais recursos cada sala possui. Em $\mathbf{C}_{m,l}$, o valor de c_{ij} é 1 se a i -ésima sala possuir o j -ésimo recurso, ou 0 se a sala não possuir o recurso.

- d) A quarta matriz, $\mathbf{D}_{n,l}$, chamada de matriz eventos-recursos, indica quais recursos são demandados por quais eventos. Em $\mathbf{D}_{n,l}$, o valor de d_{ij} é 1 se o i -ésimo evento requerer o j -ésimo recurso, ou 0 se o evento não requerer o recurso.
- e) A quinta matriz, $\mathbf{E}_{n,m}$, chamada de matriz evento-sala, indica quais eventos podem ocorrer em quais salas. Em $\mathbf{E}_{n,m}$, o valor de c_{ij} é 1 se o i -ésimo evento puder ocorrer na j -ésima sala. Através dessa matriz podemos identificar rapidamente quais salas podem acomodar quais eventos considerando-se o tamanho de cada sala e os recursos exigidos por cada evento e fornecidos pela sala.

Geralmente uma matriz é utilizada para associar cada evento a uma sala s_i e a uma lacuna de tempo t_i . Cada par (s_i, t_i) é associado a um número particular que corresponde ao evento. Se uma sala s_i estiver livre em uma lacuna de tempo t_i , então 0 é associado ao par, dizendo que não há eventos para a sala naquele momento. Ao associarmos um evento a uma sala em uma determinada lacuna de tempo, primeiro verificamos se não há nenhum outro evento agendado, se a sala estiver livre, basta aumentar o valor atual do par (s_i, t_i) correspondente em 1 unidade. Desse modo podemos garantir que nunca mais de um evento será agendado para a mesma sala na mesma lacuna de tempo, satisfazendo uma das restrições obrigatórias.

Em geral, a solução do UCTP pode ser apresentada na forma de uma lista ordenada de pares (s_i, t_i) onde o índice do par corresponde ao i -ésimo evento. Por exemplo, as salas são alocadas nas lacunas de tempo na forma:

$(2, 4), (3, 10), (1, 4), \dots, (2, 17)$

onde o 1º evento ocorre na sala 2 e na lacuna de tempo 4, a 2º na sala 3 e na lacuna 10, a 3º na sala 1 e na lacuna 4 e assim por diante.

O objetivo do UCTP é minimizar as violações das restrições opcionais das soluções viáveis e levar a zero as restrições obrigatórias. A função objetivo é ponderada pela soma das violações das restrições obrigatórias e pela soma das restrições opcionais. A função objetivo pode ser expressa na seguinte forma:

$$f(x) = C \times NVHC + NVSC$$

onde:

$NVHC$ – Número de violações das restrições obrigatórias

$NVSC$ – Número de violações das restrições opcionais

C – Constante que pondera as violações das restrições obrigatórias.

Para que uma solução que viole uma restrição obrigatória mas que satisfaça várias opcionais não seja considerada melhor que uma solução que satisfaça todas as obrigatórias mas não satisfaça várias opcionais, C deve ter um valor tal que $C > NVCS$, indicando assim que basta a não satisfação de uma única restrição obrigatória para que a solução seja considerada pior que as soluções que satisfazem todas as restrições opcionais.

Uma solução globalmente ótima será então uma solução tal que $f(x) = 0$, indicando que nenhuma restrição obrigatória ou opcional foi violada.

4.2. Modelagem Proposta

No nosso modelo, algumas modificações foram feitas em relação ao modelo proposto por Rossi-Doria et al. Nossa modelagem é dada da seguinte forma:

- a) Um evento corresponde a uma aula de uma disciplina. Cada evento está associado à disciplina a qual ele pertence. Se a disciplina possuir aulas em horário fixo, cada evento corresponderá a uma das aulas com dia e horário pré-estabelecidos da disciplina. Se a disciplina não possuir aulas fixas, cada evento terá uma duração correspondente à duração, em horas, da respectiva aula da disciplina.
- b) Um gene corresponderá a uma alocação de sala, dia e hora para um evento.
- c) A quantidade de genes é a mesma de eventos, e a alocação do i -ésimo gene se refere à alocação de sala, dia e hora para o i -ésimo evento.
- d) Uma matriz $\mathbf{A}_{e,e}$, onde e é o número de eventos, indica se dois eventos i e j são disciplinas obrigatórias do mesmo período.
- e) Uma matriz $\mathbf{B}_{e,e}$, onde e é o número de eventos, indica se dois eventos i e j são disciplinas eletivas de mesmo perfil.
- f) Uma matriz $\mathbf{C}_{e,e}$, onde e é o número de eventos, indica se dois eventos i e j tem em comum.
- g) Uma matriz $\mathbf{D}_{e,d,h}$, onde e é o número de eventos, d o número de dias na semana e h o número de aulas por dia, indica, para cada evento e , os dias d e horas h em que ele não pode ocorrer. Essa matriz é feita a partir das matrizes de cada professor que indicam quando ele não poderá dar aula.
- h) Uma matriz $\mathbf{E}_{e,d,h}$, onde e é o número de eventos, d o número de dias na semana e h o número de aulas por dia, indica, para cada evento e , os dias d e horas h em que ele pode,

mas não deve ocorrer. Essa matriz é feita a partir das matrizes de cada professor que indicam quando ele não quer dar aula.

4.3.Requisitos de Entrada e Saída

As entradas necessárias e as saídas fornecidas pelo sistema estão organizadas da seguinte forma:

4.3.1. Entradas

4.3.1.1. Professor

Dados referentes ao professor.

Variável	Tipo de Dado	Descrição
Matrícula	String	Uma matrícula que identifique o professor.
Nome	String	O nome do professor.
Trabalha manhã	Booleano	Indica se o professor trabalha no turno da manhã.
Trabalha tarde	Booleano	Indica se o professor trabalha no turno da tarde.
Trabalha noite	Booleano	Indica se o professor trabalha no turno da noite.
Horários ocupado obrigatório	Matriz de booleano	Indica, para cada lacuna de tempo dia x hora, se o professor não pode de forma alguma lecionar naquele horário.
Horários ocupado opcional	Matriz de booleano	Indica, para cada lacuna de tempo dia x hora, se o professor não deseja lecionar naquele horário, embora possa.

4.3.1.2. Disciplina

Dados de uma disciplina.

Variável	Tipo de Dado	Descrição
Código	String	O código da disciplina.
Nome	String	O nome da disciplina.
Obrigatória	Booleano	Indica se a disciplina é obrigatória ou não.
Período	Número inteiro	O número do período indicado para que a disciplina seja cursada. Desconsiderado se a disciplina for eletiva.
Perfil	String	Para disciplinas eletivas, o perfil ao qual a disciplina pertence. Ex: Engenharia de Software,

		Jogos Digitais, Sistemas Operacionais.
Horário de aula fixo	Booleano	Indica, se a disciplina tem um horário fixo. Deve ser utilizada em caso de o sistema não gerar os horários de todos os cursos de uma instituição. As disciplinas de outros centros, não-abrangidos pelo sistema, devem ter um horário fixo. Ex: Alunos de um centro têm aulas de Cálculo 1 em outro centro, sendo essas com horário já definido pelo outro centro.
Aulas fixas	Array de aula fixa	Indica os dias e horários que as aulas da disciplina ocorrem, caso as aulas possuam horário fixo.
Aulas variáveis	Array de aulas variáveis	Indica quantas aulas por semana a disciplina deve ter e qual a duração de cada aula, caso as aulas não possuam horário fixo.
Pode aulas manha	Booleano	Indica se as aulas das disciplinas podem ocorrer no turno da manhã. Desprezado se a disciplina tiver horário fixo.
Pode aulas tarde	Booleano	Indica se as aulas das disciplinas podem ocorrer no turno da tarde. Desprezado se a disciplina tiver horário fixo.
Pode aulas noite	Booleano	Indica se as aulas das disciplinas podem ocorrer no turno da noite. Desprezado se a disciplina tiver horário fixo.
Turno preferencial	Enumeração	Indica em qual dos turnos que as turmas podem ter aulas, as aulas devem ser preferencialmente lecionadas. As opções são: manhã, tarde, noite, e sem preferência. Desprezado se a disciplina tiver horário fixo.
Professores do centro	Booleano	Indica se os professores que lecionam a disciplina são do centro ou são outros professores.
Professores	Array de professor	Os professores que lecionarão a disciplina, caso esta seja lecionada por professores do centro.
Forma alocação sala	Enumeração	Indica como o sistema fará para alocar sala à disciplina. As opções são:

		<ul style="list-style-type: none"> • Sistema escolhe O sistema durante a otimização escolherá uma sala para as aulas da disciplina. • Sala fixa O usuário define em qual das salas a disciplina deverá obrigatoriamente ocorrer. • Local externo Quando a disciplina é lecionada em outro centro, deve-se marcar esta opção.
Sala	Sala	Indica a sala onde a disciplina deve ocorrer, caso a forma de alocação de sala escolhida tenha sido “sala fixa”.

4.3.1.3. Aula Variável

Indica, para uma única aula semanal de uma disciplina, qual deve ser a duração da aula em qual tipo de local ela deve ocorrer.

Variável	Tipo de Dado	Descrição
Duração	Número inteiro	Indica a duração da aula.

4.3.1.4. Aula Fixa

Indica, para uma única aula semanal de uma disciplina, qual deve ser a duração da aula em qual tipo de local ela deve ocorrer.

Variável	Tipo de Dado	Descrição
Dia da semana	Enumeração	Indica em que dia da semana a aula ocorre.
Hora início	Número inteiro	Indica em que hora do dia a aula se inicia.
Hora fim	Número inteiro	Indica em que hora do dia a aula termina.

4.3.1.5. Sala de Aula

Dados sobre a sala onde ocorrerá a aula

Variável	Tipo de Dado	Descrição
Id	String	Identificador da sala.
Capacidade	Número inteiro	Capacidade de alunos que a sala comporta.

Horários indisponível	Matriz de booleano	Indica, para cada lacuna de tempo dia x hora, se a sala está indisponível para ser utilizada naquele horário, seja em virtude de uma disciplina de outro curso que ocorre na sala ou por outros motivos, como seminários e reuniões semanais de professores.
-----------------------	--------------------	--

4.3.2. Saídas

4.3.2.1. Agendamento por Período

Um conjunto de tabelas onde as colunas de cada tabela indicam os dias da semana e as linhas indicam os horários das aulas. Cada tabela agrupa as disciplinas a serem lecionadas no respectivo período do curso que a tabela representa e outra tabela agrupa as disciplinas eletivas. Cada célula a_{ij} da tabela indica a matéria que será lecionada no dia j e no horário i e a sala onde a aula ocorrerá.

4.3.2.2. Agendamento por Sala

Um conjunto de tabelas onde as colunas de cada tabela indicam os dias da semana e as linhas indicam os horários das aulas. Cada tabela agrupa as disciplinas a serem lecionadas na respectiva sala. Cada célula a_{ij} da tabela indica a matéria que será lecionada no dia j e no horário i .

4.3.2.3. Agendamento por Professor

Um conjunto de tabelas onde as colunas de cada tabela indicam os dias da semana e as linhas indicam os horários das aulas. Cada tabela agrupa as disciplinas a serem lecionadas pelo respectivo professor. Cada célula a_{ij} da tabela indica a matéria que será lecionada no dia j e no horário i e a sala onde a aula ocorrerá.

5. Capítulo V – Algoritmo Genético de Busca Guiada

O AGBG proposto por Yang e Jat (2011) incorpora o conceito de busca local e de uma estrutura de dados, aqui denotada de MEM, para guiar a geração de filhos. O algoritmo funciona do seguinte modo: primeiramente, uma população inicial é gerada aleatoriamente através de distribuição uniforme. Depois, uma busca local é aplicada a cada indivíduo da população. Depois da inicialização da população uma estrutura MEM é construída. A MEM guarda uma lista de pares sala e lacuna de tempo (s, t) , para cada evento que tem uma penalidade 0 dos indivíduos selecionados da população. Depois disso a MEM pode ser usada para guiar a geração de filhos nas gerações seguintes. A MEM é reconstruída regularmente a cada g gerações. Em cada geração do AGBG, um ou mais filhos são gerados ou pela MEM ou pelo operador de cruzamento, dependendo de uma probabilidade p . Em seguida cada filho sofre mutação, dependendo da probabilidade m , e depois a busca local é aplicada a cada filho gerado. Por fim, são excluídos da população a quantidade os piores n indivíduos, onde n é a quantidade de filhos inserida na população. Esse processo continua até um determinado número de gerações ou até que se acha uma solução ótima.

5.1. Estrutura MEM

Na literatura, alguns pesquisadores usaram uma estrutura de dados ou memória para guardar informação útil a fim de melhorar o desempenho dos AGs e outros métodos de busca meta-heurística. No AGBG usaremos uma estrutura similar, chamada de MEM e proposta por Yang e Jat (2011), para guiar a geração de filhos. A MEM é utilizada para prover direção de exploração e exploração no espaço de busca. Ela visa aumentar a qualidade de uma solução gerada reintroduzindo parte dos melhores indivíduos das gerações passadas. A MEM consiste em uma lista de dois níveis. O primeiro nível é uma lista de eventos e o segundo uma lista de pares salas-lacunas de tempo correspondente ao evento e_i no primeiro nível. A figura abaixo representa a estrutura da MEM.

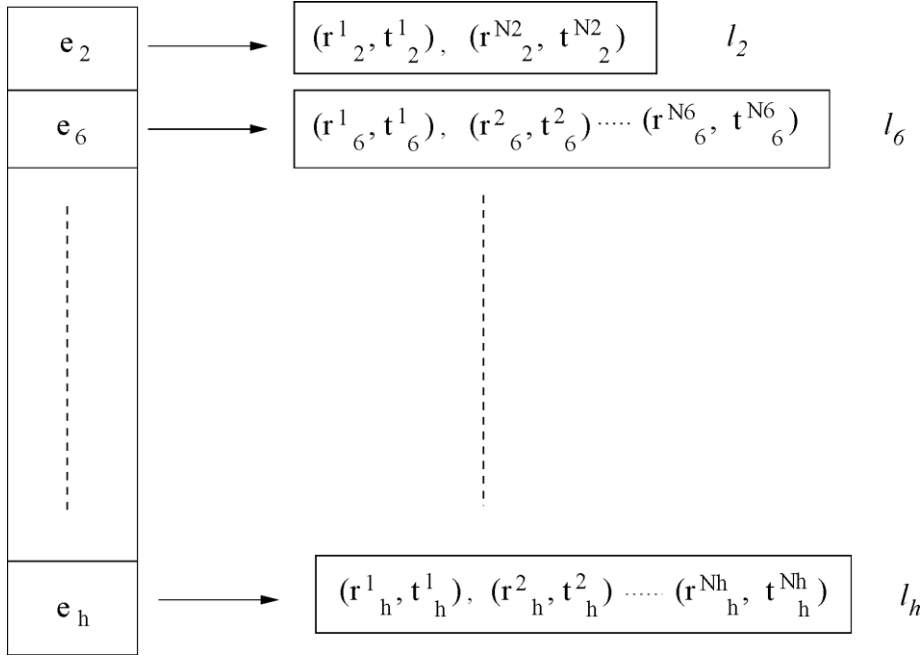


Figura 1. Representação da estrutura MEM.

A estrutura MEM é reconstruída regularmente a cada t gerações. Quando a MEM está para ser reconstruída, primeiro selecionamos os n melhores indivíduos da população P para formar um conjunto Q . Depois, para cada indivíduo $ij \in Q$, cada evento é avaliado quanto à sua penalidade. Se um evento tem penalidade 0, então guardamos a informação correspondente ao evento na MEM. Por exemplo, se um evento e_6 de um indivíduo $ij \in Q$ é alocado na sala 2 e na lacuna de tempo 13 e tem valor de penalidade 0, então adicionamos o par (2, 13) à lista l_6 . Esse procedimento é repetido para todos os indivíduos de Q . Ao final, a MEM conterá uma lista de pares salas-lacunas de tempo correspondentes aos eventos com penalidade 0 dos melhores indivíduos da população atual. O algoritmo abaixo ilustra a criação da MEM.

Algorithm 4 *ConstructMEM()*

- 1: **input** : The whole population P
 - 2: sort the population P according to the fitness of individuals
 - 3: $Q \leftarrow$ select the best $\alpha \times N$ individuals in P
 - 4: **for** each individual I_j in Q **do**
 - 5: **for** each event e_i in I_j **do**
 - 6: calculate the penalty value of event e_i from I_j
 - 7: **if** e_i is feasible (i.e., e_i has a zero penalty) **then**
 - 8: add the room and time slot pair (r_i, t_i) assigned to e_i into the list l_i
 - 9: **end if**
 - 10: **end for**
 - 11: **end for**
 - 12: **output** : The data structure MEM
-

A MEM é então utilizada para guiar a geração de filhos pelas próximas t gerações. Yang e Jat (2011) sugerem atualizar a MEM a cada t gerações em vez de atualizá-la a cada geração para balancear entre qualidade da solução e custo computacional.

5.2. Geração de Filhos

No AGBG um filho é gerado ou pela estratégia guiada de busca utilizando a MEM ou pelo operador de cruzamento com probabilidade p . Quando um filho está para ser gerado um número aleatório $x \in [0,0; 1,0]$ é gerado. Se $x < p$, o filho será gerado utilizando-se a MEM, senão, ele será gerado utilizando-se o operador de cruzamento.

A seguir descrevemos o processo para gerar um filho utilizando-se a MEM e, depois, utilizando-se o operador de cruzamento.

5.2.1. Geração de Filhos Utilizando-se a MEM

Se um filho está para ser gerado pela MEM, primeiro selecionamos um conjunto E_s de $b \times n$ eventos a serem gerados pela MEM, onde b é um valor percentual e n é o número total de eventos. De acordo com os experimentos realizados por Yang e Jat (2011), o valor ideal encontrado para b é de 30%. Com isso, temos um ganho na geração de filhos guiada pela MEM. Valores muito altos com o passar do tempo tendem a gerar filhos muito parecidos ou iguais aos indivíduos que já existem na população. Diminuindo assim a diversidade da população, fazendo com que os indivíduos tendam a convergir para um ótimo local. Depois, para cada evento $e_i \in E_s$, selecionamos aleatoriamente um par (s_{ji}, t_{ji}) , $j = 1, \dots, N_i$, da lista que corresponde ao evento e_i e associamos o par selecionado de e_i ao filho. Se há um evento em $e_i \in E_s$ que não esteja listado na MEM, alocamos aleatoriamente uma sala e lacuna de tempo das possíveis salas e lacunas de tempo para e_i ao filho. Esse processo é repetido para todos os eventos de E_s . Para os eventos que não estão presentes em E_s alocamos salas e lacunas de tempo aleatoriamente.

Algorithm 5 *GuidedSolutionConstruction()*

- 1: **input** : The *MEM* data structure
 - 2: $E_s :=$ randomly select $\beta * n$ events
 - 3: **for** each event e_i in E_s **do**
 - 4: randomly select a pair of room and time slot from the list l_i
 - 5: assign the selected pair to event e_i for the child
 - 6: **end for**
 - 7: **for** each remaining event e_i not in E_s **do**
 - 8: assign a random time slot and room to event e_i
 - 9: **end for**
 - 10: **output** : A new child generated using *MEM*
-

5.2.2. Geração de Filhos por Cruzamento

Se um filho está para ser gerado através de cruzamento, primeiro selecionamos dois indivíduos da população como pais. Yang e Jat (2011) propõem que esta seleção se dê através de torneio com tamanho de grupo 2. Então, para cada evento do filho, selecionamos aleatoriamente um pai e alocamos a lacuna de tempo correspondente para aquele evento. Finalmente, alocamos salas aos eventos em cada lacuna de tempo utilizando o algoritmo de casamento.

Algorithm 6 *Crossover()*

```
1: input : The current population
2: Select parents  $P1$  and  $P2$  by the tournament selection
3: for each event  $e_i$  of the child  $Ch$  do
4:   if  $rand(0, 1) < 0.5$  then
5:      $e_i$  of  $Ch \leftarrow$  the time slot allocated to  $e_i$  of  $P1$ 
6:   else
7:      $e_i$  of  $Ch \leftarrow$  the time slot allocated to  $e_i$  of  $P2$ 
8:   end if
9: end for
10: allocate rooms to all occupied time slots using the matching algorithm
11: output : A new child generated using Crossover()
```

5.3. Mutação

No processo de mutação cada gene do indivíduo sofre mutação com probabilidade p . O processo de mutação ocorre em duas etapas. Na primeira etapa mudamos a sala alocada ao evento. Cada evento possui uma lista pré-processada de salas nas quais o evento pode ocorrer. Para fazermos a mutação da sala, escolhemos aleatoriamente uma das salas da lista de salas que podem abrigar o evento. Dessa forma, jamais um evento será alocado em uma sala que não atenda as suas restrições. Na segunda etapa, mudamos o dia e a hora do evento. Primeiro escolhemos aleatoriamente um dos dias da semana. Em seguida, para o dia escolhido, verificamos em quantas lacunas de tempo do dia o evento pode ocorrer e qual a primeira lacuna de tempo na qual ele pode ocorrer. Então escolhemos aleatoriamente uma das lacunas na qual o evento pode ocorrer.

5.4. Busca Local

A busca local proposta por Yang e Jat (2011) é feita após a geração e mutação de um indivíduo e tem o objetivo de melhorá-lo ainda mais. A busca ocorre em três etapas, sendo as duas primeiras consideradas por Yang e Jat (2011) como a busca local 1 e a terceira a busca local 2, e é baseada em 3 estruturas de vizinhança N_1 , N_2 e N_3 , descritas a seguir:

- a) N_1 é a vizinhança definida pelo operador que move um evento de uma lacuna de tempo para outra diferente.

- b) N_2 é a vizinhança definida pelo operador que troca as lacunas de 2 eventos em lacunas de tempo diferentes.
- c) N_3 é a vizinhança definida pelo operador que permuta as lacunas de 3 eventos em lacunas de tempo diferentes.

Na primeira etapa, a busca procura determinar violações de restrições obrigatórias para cada evento, ignorando as violações das restrições opcionais. Se há violações de restrições obrigatórias para um evento, a busca tenta satisfazer a restrição aplicando movimentos nas estruturas de vizinhança N_1 , N_2 e N_3 , nesta ordem, até que uma condição de parada seja satisfeita, como a restrição ser satisfeita ou ser atingido o número máximo de tentativas de movimentos permitido. Após cada movimento, aplicamos o algoritmo de *matching* a cada lacuna do tempo afetada para resolver a perturbação na alocação das salas e fazemos uma avaliação do delta do movimento. Uma avaliação do delta significa que nós apenas calculamos as violações de restrições obrigatórias e opcionais para os eventos envolvidos em um movimento dentro de um indivíduo e comparamos a mudança do valor da função objetivo antes e depois do movimento. Se fôssemos recalcular a aptidão do indivíduo inteiro considerando todos os eventos, iríamos gastar muito mais tempo. Se não há movimentos que não foram tentados para a vizinhança de um evento, a busca passa para o próximo evento. Se após aplicar todos os movimentos de vizinhança em cada evento ainda houver algum evento com violação de uma restrição obrigatória, o algoritmo pula a segunda etapa, senão, ele segue para a segunda etapa.

Na segunda etapa, após encontrar uma solução viável, a busca lida com as restrições opcionais de um modo similar. Para cada evento, a busca tenta fazer os movimentos nas vizinhanças N_1 , N_2 e N_3 para reduzir as violações das restrições opcionais. Após cada movimento aplicamos o algoritmo de *matching* a cada lacuna do tempo afetada para tentar resolver a perturbação na alocação das salas e fazemos uma avaliação do delta do movimento.

Na terceira etapa buscamos por lacunas de tempo com alta penalidade, pois isso indica que ela pode ter vários eventos envolvidos em violações de restrições obrigatórias e opcionais. Inicialmente, a busca seleciona aleatoriamente um percentual fixo de lacunas de tempo. Em vez de escolher a pior de todas as lacunas, algumas lacunas são selecionadas aleatoriamente e a pior delas é escolhida. Isso se deve ao fato de que para cada lacuna selecionada deve-se calcular a penalidade da lacuna, o que demanda muito tempo. Selecionando algumas lacunas aleatórias balanceamos entre qualidade e tempo computacional. De acordo com os experimentos realizados por Yang e Jat (2011), a seleção aleatória de um percentual de 20% das lacunas mostrou bons resultados. Depois, a busca calcula a

penalidade para cada lacuna, somando as penalidades de cada evento que está alocada na lacuna, e escolhe a lacuna com a maior penalidade. A busca tenta então mover cada evento da lacuna na vizinhança N_1 . Se todos os movimentos a serem aplicados nos eventos das lacunas reduzirem a penalidade da lacuna, aplicamos os movimentos, senão, eles são descartados. Desse modo a busca pode não apenas averiguar a pior lacuna como também reduzir a penalidade dela movendo os eventos para outras lacunas de tempo.

Quando a busca é encerrada, muito possivelmente teremos um indivíduo melhorado. O algoritmo de busca está representado na figura abaixo:

Algorithm 2 Local Search Scheme 1 (LS1)

```

1: input : Individual I from the population
2: for each event  $e_i \in E$  do
3:   if event  $e_i$  is infeasible then
4:     if there is untried move left then
5:       calculate the moves: first N1, then N2 if N1 fails,
        and finally N3 if N1 and N2 fail
6:       apply the matching algorithm to the time slots
        affected by the move to allocate rooms for events
7:       delta evaluate the result of the move
8:       if moves reduce hard constraints violation then
9:         make the moves and go to line 4
10:      end if
11:    end if
12:  end if
13: end for
14: if no any hard constraints remain then
15:   for each event  $e_i \in E$  do
16:     if event  $e_i$  has soft constraint violation then
17:       if there is untried move left then
18:         calculate the moves: first N1, then N2 if N1 fails,
        and finally N3 if N1 and N2 fail
19:         apply the matching algorithm to the time slots
        affected by the move to allocate rooms for
        events
20:         delta evaluate the result of the move
21:         if moves reduce soft constraints violation then
22:           make the moves and go to line 17
23:         end if
24:       end if
25:     end if
26:   end for
27: end if
28: output : A possibly improved individual I

```

Algorithm 7 Local Search Scheme 2 (LS2)

```
1: input : Individual I after LS1 is applied
2:  $S :=$  randomly select a preset percentage of time slots from the
   total time slots of  $T$ 
3: for each time slot  $t_i \in S$  do
4:   for each event  $j$  in time slot  $t_i$  do
5:     calculate the penalty value of event  $j$ 
6:   end for
7:   sum the total penalty value of events in time slot  $t_i$ 
8: end for
9: select the time slot  $w_t$  with the biggest penalty value from  $S$ 
10: for each event  $i$  in  $w_t$  do
11:   calculate a move of event  $i$  in the neighbourhood structure N1
12:   apply the matching algorithm to the time slots affected by the
     move
13:   compute the penalty of event  $i$  and delta evaluate the result
14: end for
15: if all the moves together reduce hard or soft constraint violations
     then
16:   apply the moves
17: else
18:   delete the moves
19: end if
20: output : A possibly improved individual I
```

6. Capítulo VI – Resultados Obtidos

Foram realizados vários experimentos, totalizando mais de 90 simulações onde variamos vários parâmetros. Nesta seção mostraremos como está o horário atual e quais os melhores resultados obtidos considerando dois cenários: o modelo atual, onde algumas disciplinas são dadas em outros centros e outras são lecionadas em conjunto com o curso de Ciências da Computação, e um modelo onde todas as disciplinas são lecionadas no CIn e o horário de todas, inclusive as que ocorrem em conjunto com o curso de Ciências da Computação, tem o seu horário definido pelo sistema.

Em todas as tabelas apresentadas a seguir, para as disciplinas de obrigatórias, um item na cor vermelha indica que aquele item violou uma restrição obrigatória, um item na cor roxa indica que aquele item violou uma restrição opcional, e um item na cor preta indica que ele não violou restrição alguma. Para as disciplinas eletivas, um item na cor vermelha indica que aquele item violou uma restrição obrigatória, um item na cor roxa indica que aquele item violou uma restrição opcional. Caso um item de uma disciplina eletiva não tenha violado restrição alguma, o item é colorido de acordo com o perfil da disciplina eletiva. Itens de um mesmo perfil estão em uma mesma cor.

6.1. Horário Atual

6.1.1. Agendamento

Disciplinas do 1º período

	Segunda	Terça	Quarta	Quinta	Sexta
8:00hs			(Local externo) MA026 - Cálculo Diferencial e Integral 1		
9:00hs			(Local externo) MA026 - Cálculo Diferencial e Integral 1		
10:00hs					(Local externo) MA026 - Cálculo Diferencial e Integral 1
11:00hs					(Local externo) MA026 - Cálculo

					Diferencial e Integral 1
12:00hs					
13:00hs	(D004) IF669 - Introdução à Programação (EC)	(D004) IF668 - Introdução à Computação (EC)	(D004) MA531 - Álgebra Vetorial e Linear para Computação (EC)	(D004) IF669 - Introdução à Programação (EC)	(D004) IF670 - Matemática Discreta
14:00hs	(D004) IF669 - Introdução à Programação (EC)	(D004) IF668 - Introdução à Computação (EC)	(D004) MA531 - Álgebra Vetorial e Linear para Computação (EC)	(D004) IF669 - Introdução à Programação (EC)	(D004) IF670 - Matemática Discreta
15:00hs		(D004) IF669 - Introdução à Programação (EC)	(D004) IF670 - Matemática Discreta		(D004) MA531 - Álgebra Vetorial e Linear para Computação (EC)
16:00hs		(D004) IF669 - Introdução à Programação (EC)	(D004) IF670 - Matemática Discreta		(D004) MA531 - Álgebra Vetorial e Linear para Computação (EC)
17:00hs					

Disciplinas do 2º período

	Segunda	Terça	Quarta	Quinta	Sexta
8:00hs		(D003) IF672 - Algoritmos (EC)	(D003) IF675 - Sistemas Digitais (EC)	(D003) IF673 - Lógica (EC)	(Local externo) FI006 - Física Geral 1
9:00hs		(D003) IF672 - Algoritmos (EC)	(D003) IF675 - Sistemas Digitais (EC)	(D003) IF673 - Lógica (EC)	(Local externo) FI006 - Física Geral 1
10:00hs	(Local externo) FI006 - Física Geral 1	(D003) IF675 - Sistemas Digitais (EC)		(D003) IF672 - Algoritmos (EC)	
11:00hs	(Local externo) FI006 - Física Geral 1	(D003) IF675 - Sistemas Digitais (EC)		(D003) IF672 - Algoritmos (EC)	
12:00hs					
13:00hs	(D003) IF673 - Lógica (EC)		(Local externo) MA027 - Cálculo Diferencial e Integral 2		(Local externo) MA027 - Cálculo Diferencial e Integral 2
14:00hs	(D003) IF673 - Lógica (EC)		(Local externo) MA027 - Cálculo Diferencial e Integral 2		(Local externo) MA027 - Cálculo Diferencial e Integral 2
15:00hs					
16:00hs					
17:00hs					

Disciplinas do 3º período

	Segunda	Terça	Quarta	Quinta	Sexta
8:00hs	(Local externo) MA028 - Cálculo Diferencial e Integral 3				
9:00hs	(Local externo) MA028 - Cálculo Diferencial e Integral 3				
10:00hs			(Local externo) MA028 - Cálculo Diferencial e Integral 3		
11:00hs			(Local externo) MA028 - Cálculo Diferencial e Integral 3		
12:00hs					
13:00hs	(Local externo) FI007 - Física Geral 2	(D005) IF679 - Informática e Sociedade (EC)		(D005) IF677 - Infra-estrutura de Software (EC)	(D001) IF674 - Infra- estrutura de Hardware (EC)
14:00hs	(Local externo) FI007 - Física Geral 2	(D005) IF679 - Informática e Sociedade (EC)		(D005) IF677 - Infra-estrutura de Software (EC)	(D001) IF674 - Infra- estrutura de Hardware (EC)
15:00hs	(D001) IF674 - Infra-estrutura de Hardware (EC)	(D005) IF677 - Infra-estrutura de Software (EC)	(Local externo) FI007 - Física Geral 2	(D005) IF679 - Informática e Sociedade (EC)	
16:00hs	(D001) IF674 - Infra-estrutura de Hardware (EC)	(D005) IF677 - Infra-estrutura de Software (EC)	(Local externo) FI007 - Física Geral 2	(D005) IF679 - Informática e Sociedade (EC)	
17:00hs					

Disciplinas do 4º período

	Segunda	Terça	Quarta	Quinta	Sexta
8:00hs	(D003) IF815 - Métodos Numéricos	(D220) IF685 - Gerenciamento de Dados e Informação (EC)	(Local externo) FI108 - Física Geral 3		
9:00hs	(D003) IF815 - Métodos Numéricos	(D220) IF685 - Gerenciamento de Dados e Informação (EC)	(Local externo) FI108 - Física Geral 3		
10:00hs	(Local externo) FI108 - Física Geral 3		(D003) IF815 - Métodos Numéricos	(D220) IF685 - Gerenciamento de Dados e Informação (EC)	
11:00hs	(Local externo) FI108 - Física Geral 3		(D003) IF815 - Métodos Numéricos	(D220) IF685 - Gerenciamento de Dados e Informação (EC)	
12:00hs					
13:00hs		(D001) ET586 - Probabilidade e Estatística		(D001) IF817 - Interface Hardware-Software	
14:00hs		(D001) ET586 - Probabilidade e Estatística		(D001) IF817 - Interface Hardware-Software	
15:00hs		(D001) IF817 - Interface Hardware-Software		(D001) ET586 - Probabilidade e Estatística	
16:00hs		(D001) IF817 - Interface Hardware-Software		(D001) ET586 - Probabilidade e Estatística	
17:00hs					

Disciplinas do 5º período

	Segunda	Terça	Quarta	Quinta	Sexta
8:00hs					
9:00hs					
10:00hs					
11:00hs					
12:00hs					
13:00hs	(D002) ES413 - Sinais e Sistemas	(Local externo) EL215 - Circuitos Elétricos	(D002) IF678 - Infra-estrutura de Comunicação	(Local externo) EL215 - Circuitos Elétricos	(D002) IF690 - História e Futuro da Computação
14:00hs	(D002) ES413 - Sinais e Sistemas	(Local externo) EL215 - Circuitos Elétricos	(D002) IF678 - Infra-estrutura de Comunicação	(Local externo) EL215 - Circuitos Elétricos	(D002) IF690 - História e Futuro da Computação
15:00hs		(D003) IF686 - Paradigma de Linguagens Computacionais (EC)	(D002) ES413 - Sinais e Sistemas	(D003) IF686 - Paradigma de Linguagens Computacionais (EC)	(D002) IF678 - Infra-estrutura de Comunicação
16:00hs		(D003) IF686 - Paradigma de Linguagens Computacionais (EC)	(D002) ES413 - Sinais e Sistemas	(D003) IF686 - Paradigma de Linguagens Computacionais (EC)	(D002) IF678 - Infra-estrutura de Comunicação
17:00hs					

Disciplinas do 6º período

	Segunda	Terça	Quarta	Quinta	Sexta
8:00hs	(D002) IF682 - Engenharia de Software e Sistemas		(D002) IF682 - Engenharia de Software e Sistemas		(D005) IF689 - Informática Teórica
9:00hs	(D002) IF682 - Engenharia de Software e Sistemas		(D002) IF682 - Engenharia de Software e Sistemas	(Local externo) ES203 - Eletromagnetismo 1	(D005) IF689 - Informática Teórica
10:00hs	(Local externo) ES203 - Eletromagnetismo 1	(D005) IF689 - Informática Teórica	(Local externo) ES232 - Introdução aos Dispositivos Semicondutores	(Local externo) ES203 - Eletromagnetismo 1	(Local externo) ES232 - Introdução aos Dispositivos Semicondutores
11:00hs	(Local externo) ES203 - Eletromagnetismo 1	(D005) IF689 - Informática Teórica	(Local externo) ES232 - Introdução aos Dispositivos Semicondutores		(Local externo) ES232 - Introdução aos Dispositivos Semicondutores
12:00hs	(Local externo) ES203 - Eletromagnetismo 1				
13:00hs	(D005) LE530 - Inglês para Computação				
14:00hs	(D005) LE530 - Inglês para Computação				
15:00hs			(D005) LE530 - Inglês para Computação		
16:00hs			(D005) LE530 - Inglês para Computação		
17:00hs					

Disciplinas do 7º período

	Segunda	Terça	Quarta	Quinta	Sexta
8:00hs			(D005) IF684 - Sistemas Inteligentes		(D002) ES238 - Eletrônica 1
9:00hs	(Local externo) ES344 - Princípios de Comunicação		(D005) IF684 - Sistemas Inteligentes	(Local externo) ES344 - Princípios de Comunicação	(D002) ES238 - Eletrônica 1
10:00hs	(Local externo) ES344 - Princípios de Comunicação		(D002) ES238 - Eletrônica 1	(Local externo) ES344 - Princípios de Comunicação	(D005) IF684 - Sistemas Inteligentes
11:00hs			(D002) ES238 - Eletrônica 1	(Local externo) ES344 - Princípios de Comunicação	(D005) IF684 - Sistemas Inteligentes
12:00hs					
13:00hs			(D001) IF688 - Teoria e Implementação de Linguagens Computacionais		
14:00hs			(D001) IF688 - Teoria e Implementação de Linguagens Computacionais		
15:00hs		(Local externo) ES414 - Servomecanismo		(Local externo) ES414 - Servomecanismo	(D001) IF688 - Teoria e Implementação de Linguagens Computacionais
16:00hs		(Local externo) ES414 - Servomecanismo		(Local externo) ES414 - Servomecanismo	(D001) IF688 - Teoria e Implementação de Linguagens Computacionais
17:00hs					

Disciplinas do 8º período

	Segunda	Terça	Quarta	Quinta	Sexta
8:00hs	(D220) IF819 - Projeto de Co-desenvolvimento Hardware-software		(D220) IF819 - Projeto de Co-desenvolvimento Hardware-software		
9:00hs	(D220) IF819 - Projeto de Co-desenvolvimento Hardware-software		(D220) IF819 - Projeto de Co-desenvolvimento Hardware-software		
10:00hs					
11:00hs					
12:00hs					
13:00hs	(D001) IF676 - Metodologia e Expressão Técnico-científica				
14:00hs	(D001) IF676 - Metodologia e Expressão Técnico-científica				
15:00hs			(D001) IF676 - Metodologia e Expressão Técnico-científica		
16:00hs			(D001) IF676 - Metodologia e Expressão Técnico-científica		
17:00hs					

Disciplinas eletivas

	Segunda	Terça	Quarta	Quinta	Sexta
8:00hs	(Anfiteatro) IF781 - Empreendimentos em Informática (D004) IF738 - Redes de Computadores	(D218) IF702 - Redes Neurais	(Auditório) IF722 - Tópicos Avançados em Engenharia de Software	(D001) IF717 - Planejamento e Gerenciamento de Projetos	(D002) IF735 - Tópicos Avançados em Arquitetura de Computadores
9:00hs	(Anfiteatro) IF781 - Empreendimentos em Informática (D004) IF738 - Redes de Computadores	(D218) IF702 - Redes Neurais	(Auditório) IF722 - Tópicos Avançados em Engenharia de Software	(D001) IF717 - Planejamento e Gerenciamento de Projetos	(D002) IF735 - Tópicos Avançados em Arquitetura de Computadores
10:00hs	(Anfiteatro) IF781 - Empreendimentos em Informática (D218) IF700 - Percepção Computacional e Reconhecimento de Padrões (D001) IF717 - Planejamento e Gerenciamento de Projetos (Auditório) IF722 - Tópicos Avançados em Engenharia de Software		(D218) IF700 - Percepção Computacional e Reconhecimento de Padrões (Anfiteatro) IF751 - Processamento de Imagens (D004) IF738 - Redes de Computadores	(D218) IF702 - Redes Neurais	(D002) IF735 - Tópicos Avançados em Arquitetura de Computadores
11:00hs	(Anfiteatro) IF781 - Empreendimentos em Informática (D218) IF700 -		(D218) IF700 - Percepção Computacional e Reconhecimento de Padrões	(D218) IF702 - Redes Neurais	(D002) IF735 - Tópicos Avançados em Arquitetura de Computadores

	<p>Percepção Computacional e Reconhecimento de Padrões</p> <p>(D001) IF717 - Planejamento e Gerenciamento de Projetos</p> <p>(Auditório) IF722 - Tópicos Avançados em Engenharia de Software</p>		<p>(Anfiteatro) IF751 - Processamento de Imagens</p> <p>(D004) IF738 - Redes de Computadores</p>		
12:00hs	(D004) IF778 - Seminário em Informática Teórica	(D224) IF707 - Seminário em Inteligência Artificial	(D218) IF778 - Seminário em Informática Teórica	(D222) IF707 - Seminário em Inteligência Artificial	
13:00hs	<p>(D220) IF730 - Sistemas de Tempo Real</p> <p>(D218) IF768 - Teoria dos Grafos</p> <p>(Local externo) IF775 - Tópicos Avançados em Algoritmos</p>	<p>(Auditório) IF705 - Automação Inteligente</p> <p>(D220) IF696 - Integração de Dados e Warehousing</p>	(D220) IF814 - Metodologia de Hardware/Software e Co-design	(D218) IF716 - Especificação de Requisitos e Validação de Sistemas	(Auditório) IF705 - Automação Inteligente
14:00hs	<p>(D220) IF730 - Sistemas de Tempo Real</p> <p>(D218) IF768 - Teoria dos Grafos</p> <p>(Local externo) IF775 - Tópicos Avançados em Algoritmos</p>	<p>(Auditório) IF705 - Automação Inteligente</p> <p>(D220) IF696 - Integração de Dados e Warehousing</p>	(D220) IF814 - Metodologia de Hardware/Software e Co-design	(D218) IF716 - Especificação de Requisitos e Validação de Sistemas	(Auditório) IF705 - Automação Inteligente
15:00hs	(D220) IF730 - Sistemas de Tempo Real	(D218) IF716 - Especificação	(D218) IF768 - Teoria dos Grafos	(D220) IF696 - Integração de Dados e	(D220) IF814 - Metodologia de Hardware/Software

		o de Requisitos e Validação de Sistemas	(Local externo) IF775 - Tópicos Avançados em Algoritmos	Warehousing	e Co-desing
16:00h s	(D220) IF730 - Sistemas de Tempo Real	(D218) IF716 - Especificaçã o de Requisitos e Validação de Sistemas	(D218) IF768 - Teoria dos Grafos (Local externo) IF775 - Tópicos Avançados em Algoritmos	(D220) IF696 - Integração de Dados e Warehousing	(D220) IF814 - Metodologia de Hardware/Softwar e Co-desing
17:00h s	(Anfiteatro) IF751 - Processamento de Imagens				
18:00h s	(Anfiteatro) IF751 - Processamento de Imagens				

6.2. Resultados Obtidos

6.2.1. Penalidades

Para os resultados, adotamos as seguintes penalidades, sendo a penalidade para violação das restrições obrigatórias configuradas independentemente para cada experimento. Os ajustes destes valores de penalidade foram feitos por tentativa e erro.

Tabela 3 - Penalidade das restrições violadas.

Restrição violada	Penalidade
Restrição obrigatória, qualquer que seja.	Configurável.
Evento alocado fora do turno preferencial, quando houver.	+4
Eventos de disciplinas eletivas de mesmo perfil ocorrendo ao mesmo tempo.	+3
Eventos de uma mesma disciplina são alocados em dias seguidos, sem folga mínima de 1 dia.	+2
Eventos de uma mesma disciplina não alternam o horário das aulas.	+1
Um professor é alocado em um horário que ele não gostaria de dar aula.	+1

6.2.2. Considerando Disciplinas Externas com Horários Fixos

Tempo gasto para gerar os resultados: 0:0:9hs

6.2.2.1. Parâmetros utilizados

Número de gerações: 1000

Número de indivíduos: 20

Número de gerações com soluções aceitáveis para parar: 10

Penalidade para violação de restrição obrigatória: 1000

Percentual da população que compete no torneio: 0,25

Probabilidade de cruzamento: 0,5

Probabilidade de mutação: 0,1

Intervalo de atualização da MEM: 20

Percentual de indivíduos na criação da MEM: 0,2

Percentual de eventos da MEM inseridos num filho gerado: 0,3

6.2.2.2. Restrições Violadas

Restrições Obrigatórias (0)

Restrições Opcionais (11)

A disciplina Cálculo Diferencial e Integral 2 (MA027) não está alternando o horário das aulas.

A disciplina Circuitos Elétricos (EL215) não está alternando o horário das aulas.

A disciplina Eletromagnetismo 1 (ES203) não está alternando o horário das aulas.

A disciplina Engenharia de Software e Sistemas (IF682) não está alternando o horário das aulas.

A disciplina Introdução à Programação (EC) (IF669) está com aulas em dias seguidos.

A disciplina Introdução à Programação (EC) (IF669) não está alternando o horário das aulas.

A disciplina Introdução aos Dispositivos Semicondutores (ES232) não está alternando o horário das aulas.

A disciplina Princípios de Comunicação (ES344) não está alternando o horário das aulas.

A disciplina Seminário em Informática Teórica (IF778) não está alternando o horário das aulas.

A disciplina Seminário em Inteligência Artificial (IF707) não está alternando o horário das aulas.

A disciplina Servomecanismo (ES414) não está alternando o horário das aulas.

6.2.2.3. Agendamento

Disciplinas do 1º período

	Segunda	Terça	Quarta	Quinta	Sexta
8:00hs		(D005) IF670 - Matemática Discreta	(Local externo) MA026 - Cálculo Diferencial e Integral 1	(D003) IF669 - Introdução à Programação (EC)	(D002) IF669 - Introdução à Programação (EC)
9:00hs	(D001) MA531 - Álgebra Vetorial e Linear para Computação (EC)	(D005) IF670 - Matemática Discreta	(Local externo) MA026 - Cálculo Diferencial e Integral 1	(D003) IF669 - Introdução à Programação (EC)	(D002) IF669 - Introdução à Programação (EC)
10:00hs	(D001) MA531 - Álgebra Vetorial e Linear para Computação (EC)	(D003) IF668 - Introdução à Computação (EC)		(Anfiteatro) IF670 - Matemática Discreta	(Local externo) MA026 - Cálculo Diferencial e Integral 1
11:00hs	(D004) IF669 - Introdução à Programação (EC)	(D003) IF668 - Introdução à Computação (EC)	(D004) MA531 - Álgebra Vetorial e Linear para Computação (EC)	(Anfiteatro) IF670 - Matemática Discreta	(Local externo) MA026 - Cálculo Diferencial e Integral 1
12:00hs	(D004) IF669 - Introdução à Programação (EC)		(D004) MA531 - Álgebra Vetorial e Linear para Computação (EC)		
13:00hs					
14:00hs					
15:00hs					
16:00hs					
17:00hs					

Disciplinas do 2º período

	Segunda	Terça	Quarta	Quinta	Sexta
8:00hs					(Local externo) FI006 - Física Geral 1
9:00hs					(Local externo) FI006 - Física Geral 1
10:00hs	(Local externo) FI006 - Física Geral 1				
11:00hs	(Local externo) FI006 - Física Geral 1				
12:00hs					
13:00hs	(D004) IF675 - Sistemas Digitais (EC)	(D001) IF673 - Lógica (EC)	(Local externo) MA027 - Cálculo Diferencial e Integral 2	(D001) IF672 - Algoritmos (EC)	(Local externo) MA027 - Cálculo Diferencial e Integral 2
14:00hs	(D004) IF675 - Sistemas Digitais (EC)	(D001) IF673 - Lógica (EC)	(Local externo) MA027 - Cálculo Diferencial e Integral 2	(D001) IF672 - Algoritmos (EC)	(Local externo) MA027 - Cálculo Diferencial e Integral 2
15:00hs	(D004) IF672 - Algoritmos (EC)		(D002) IF675 - Sistemas Digitais (EC)		(D002) IF673 - Lógica (EC)
16:00hs	(D004) IF672 - Algoritmos (EC)		(D002) IF675 - Sistemas Digitais (EC)		(D002) IF673 - Lógica (EC)
17:00hs					

Disciplinas do 3º período

	Segunda	Terça	Quarta	Quinta	Sexta
8:00hs	(Local externo) MA028 - Cálculo Diferencial e Integral 3	(D003) IF677 - Infra-estrutura de Software (EC)	(D003) IF674 - Infra-estrutura de Hardware (EC)	(D001) IF679 - Informática e Sociedade (EC)	
9:00hs	(Local externo) MA028 - Cálculo Diferencial e Integral 3	(D003) IF677 - Infra-estrutura de Software (EC)	(D003) IF674 - Infra-estrutura de Hardware (EC)	(D001) IF679 - Informática e Sociedade (EC)	
10:00hs	(D003) IF679 - Informática e Sociedade (EC)		(Local externo) MA028 - Cálculo Diferencial e Integral 3	(D003) IF677 - Infra-estrutura de Software (EC)	
11:00hs	(D003) IF679 - Informática e Sociedade (EC)		(Local externo) MA028 - Cálculo Diferencial e Integral 3	(D003) IF677 - Infra-estrutura de Software (EC)	(D005) IF674 - Infra-estrutura de Hardware (EC)
12:00hs					(D005) IF674 - Infra-estrutura de Hardware (EC)
13:00hs	(Local externo) FI007 - Física Geral 2				
14:00hs	(Local externo) FI007 - Física Geral 2				
15:00hs			(Local externo) FI007 - Física Geral 2		
16:00hs			(Local externo) FI007 - Física Geral 2		
17:00hs					

Disciplinas do 4º período

	Segunda	Terça	Quarta	Quinta	Sexta
8:00hs			(Local externo) FI108 - Física Geral 3		
9:00hs			(Local externo) FI108 - Física Geral 3		
10:00hs	(Local externo) FI108 - Física Geral 3				
11:00hs	(Local externo) FI108 - Física Geral 3				
12:00hs					
13:00hs	(D002) ET586 - Probabilidade e Estatística	(D004) IF817 - Interface Hardware-Software		(D004) IF815 - Métodos Numéricos	(D001) IF685 - Gerenciamento de Dados e Informação (EC)
14:00hs	(D002) ET586 - Probabilidade e Estatística	(D004) IF817 - Interface Hardware-Software		(D004) IF815 - Métodos Numéricos	(D001) IF685 - Gerenciamento de Dados e Informação (EC)
15:00hs	(D002) IF815 - Métodos Numéricos	(D001) IF685 - Gerenciamento de Dados e Informação (EC)		(D003) IF817 - Interface Hardware-Software	(D005) ET586 - Probabilidade e Estatística
16:00hs	(D002) IF815 - Métodos Numéricos	(D001) IF685 - Gerenciamento de Dados e Informação (EC)		(D003) IF817 - Interface Hardware-Software	(D005) ET586 - Probabilidade e Estatística
17:00hs					

Disciplinas do 5º período

	Segunda	Terça	Quarta	Quinta	Sexta
8:00hs	(D003) IF678 - Infra-estrutura de Comunicação		(D005) ES413 - Sinais e Sistemas	(D004) IF686 - Paradigma de Linguagens Computacionais (EC)	
9:00hs	(D003) IF678 - Infra-estrutura de Comunicação		(D005) ES413 - Sinais e Sistemas	(D004) IF686 - Paradigma de Linguagens Computacionais (EC)	
10:00hs		(D004) IF686 - Paradigma de Linguagens Computacionais (EC)	(D002) IF678 - Infra-estrutura de Comunicação		(D002) ES413 - Sinais e Sistemas
11:00hs	(D001) IF690 - História e Futuro da Computação	(D004) IF686 - Paradigma de Linguagens Computacionais (EC)	(D002) IF678 - Infra-estrutura de Comunicação		(D002) ES413 - Sinais e Sistemas
12:00hs	(D001) IF690 - História e Futuro da Computação				
13:00hs		(Local externo) EL215 - Circuitos Elétricos		(Local externo) EL215 - Circuitos Elétricos	
14:00hs		(Local externo) EL215 - Circuitos Elétricos		(Local externo) EL215 - Circuitos Elétricos	
15:00hs					
16:00hs					
17:00hs					

Disciplinas do 6º período

	Segunda	Terça	Quarta	Quinta	Sexta
8:00hs	(D002) IF682 - Engenharia de Software e Sistemas		(D002) IF682 - Engenharia de Software e Sistemas		(D005) IF689 - Informática Teórica
9:00hs	(D002) IF682 - Engenharia de Software e Sistemas		(D002) IF682 - Engenharia de Software e Sistemas	(Local externo) ES203 - Eletromagnetismo 1	(D005) IF689 - Informática Teórica
10:00hs	(Local externo) ES203 - Eletromagnetismo 1	(D005) IF689 - Informática Teórica	(Local externo) ES232 - Introdução aos Dispositivos Semicondutores	(Local externo) ES203 - Eletromagnetismo 1	(Local externo) ES232 - Introdução aos Dispositivos Semicondutores
11:00hs	(Local externo) ES203 - Eletromagnetismo 1	(D005) IF689 - Informática Teórica	(Local externo) ES232 - Introdução aos Dispositivos Semicondutores		(Local externo) ES232 - Introdução aos Dispositivos Semicondutores
12:00hs	(Local externo) ES203 - Eletromagnetismo 1				
13:00hs	(D005) LE530 - Inglês para Computação				
14:00hs	(D005) LE530 - Inglês para Computação				
15:00hs			(D005) LE530 - Inglês para Computação		
16:00hs			(D005) LE530 - Inglês para Computação		
17:00hs					

Disciplinas do 7º período

	Segunda	Terça	Quarta	Quinta	Sexta
8:00hs		(D218) IF684 - Sistemas Inteligentes			
9:00hs	(Local externo) ES344 - Princípios de Comunicação	(D218) IF684 - Sistemas Inteligentes		(Local externo) ES344 - Princípios de Comunicação	
10:00hs	(Local externo) ES344 - Princípios de Comunicação			(Local externo) ES344 - Princípios de Comunicação	
11:00hs	(B020) ES238 - Eletrônica 1			(Local externo) ES344 - Princípios de Comunicação	(D222) IF684 - Sistemas Inteligentes
12:00hs	(B020) ES238 - Eletrônica 1				(D222) IF684 - Sistemas Inteligentes
13:00hs			(D001) IF688 - Teoria e Implementação de Linguagens Computacionais	(B020) ES238 - Eletrônica 1	
14:00hs			(D001) IF688 - Teoria e Implementação de Linguagens Computacionais	(B020) ES238 - Eletrônica 1	
15:00hs		(Local externo) ES414 - Servomecanismo		(Local externo) ES414 - Servomecanismo	(D001) IF688 - Teoria e Implementação de Linguagens Computacionais
16:00hs		(Local externo) ES414 - Servomecanismo		(Local externo) ES414 - Servomecanismo	(D001) IF688 - Teoria e Implementação de Linguagens Computacionais

17:00hs					
---------	--	--	--	--	--

Disciplinas do 8º período

	Segunda	Terça	Quarta	Quinta	Sexta
8:00hs					
9:00hs					
10:00hs					
11:00hs					
12:00hs					
13:00hs	(D001) IF676 - Metodologia e Expressão Técnico-científica		(B020) IF819 - Projeto de Co-desenvolvimento Hardware-software		
14:00hs	(D001) IF676 - Metodologia e Expressão Técnico-científica		(B020) IF819 - Projeto de Co-desenvolvimento Hardware-software		
15:00hs	(B020) IF819 - Projeto de Co-desenvolvimento Hardware-software		(D001) IF676 - Metodologia e Expressão Técnico-científica		
16:00hs	(B020) IF819 - Projeto de Co-desenvolvimento Hardware-software		(D001) IF676 - Metodologia e Expressão Técnico-científica		
17:00hs					

Disciplinas eletivas

	Segunda	Terça	Quarta	Quinta	Sexta
8:00hs	<p>(B020) IF705 - Automação Inteligente</p> <p>(Anfiteatro) IF781 - Empreendimentos em Informática</p> <p>(D220) IF717 - Planejamento e Gerenciamento de Projetos</p> <p>(D218) IF738 - Redes de Computadores</p>	<p>(B020) IF814 - Metodologia de Hardware/Software e Co-designing</p>	<p>(B020) IF716 - Especificação de Requisitos e Validação de Sistemas</p>	<p>(B020) IF696 - Integração de Dados e Warehousing</p> <p>(D218) IF700 - Percepção Computacional e Reconhecimento de Padrões</p>	<p>(B020) IF768 - Teoria dos Grafos</p> <p>(D003) IF735 - Tópicos Avançados em Arquitetura de Computadores</p>
9:00hs	<p>(B020) IF705 - Automação Inteligente</p> <p>(Anfiteatro) IF781 - Empreendimentos em Informática</p> <p>(D220) IF717 - Planejamento e Gerenciamento de Projetos</p> <p>(D218) IF738 - Redes de Computadores</p>	<p>(B020) IF814 - Metodologia de Hardware/Software e Co-designing</p> <p>(D226) IF722 - Tópicos Avançados em Engenharia de Software</p>	<p>(B020) IF716 - Especificação de Requisitos e Validação de Sistemas</p>	<p>(B020) IF696 - Integração de Dados e Warehousing</p> <p>(D218) IF700 - Percepção Computacional e Reconhecimento de Padrões</p>	<p>(B020) IF768 - Teoria dos Grafos</p> <p>(D003) IF735 - Tópicos Avançados em Arquitetura de Computadores</p>
10:00hs	<p>(Anfiteatro) IF781 - Empreendimentos em Informática</p>	<p>(D226) IF722 - Tópicos Avançados em Engenharia de Software</p>	<p>(B020) IF775 - Tópicos Avançados em Algoritmos</p>	<p>(B020) IF814 - Metodologia de Hardware/Software e Co-designing</p>	<p>(B020) IF717 - Planejamento e Gerenciamento de Projetos</p> <p>(D003) IF735 - Tópicos</p>

					Avançados em Arquitetura de Computadores
11:00hs	(Anfiteatro) IF781 - Empreendimentos em Informática	(B020) IF700 - Percepção Computacional e Reconhecimento de Padrões	(B020) IF775 - Tópicos Avançados em Algoritmos	(B020) IF814 - Metodologia de Hardware/Software e Co-design (D218) IF751 - Processamento de Imagens (D220) IF738 - Redes de Computadores	(B020) IF717 - Planejamento e Gerenciamento de Projetos (D003) IF735 - Tópicos Avançados em Arquitetura de Computadores
12:00hs	(D003) IF778 - Seminário em Informática Teórica	(B020) IF700 - Percepção Computacional e Reconhecimento de Padrões (D003) IF707 - Seminário em Inteligência Artificial	(D003) IF778 - Seminário em Informática Teórica	(D218) IF751 - Processamento de Imagens (D220) IF738 - Redes de Computadores (D003) IF707 - Seminário em Inteligência Artificial	
13:00hs	(B020) IF696 - Integração de Dados e Warehousing (D218) IF702 - Redes Neurais (D003) IF730 - Sistemas de Tempo Real		(D218) IF705 - Automação Inteligente		(D220) IF775 - Tópicos Avançados em Algoritmos
14:00hs	(B020) IF696 - Integração de Dados e Warehousing (D218) IF702 -		(D218) IF705 - Automação Inteligente		(D220) IF775 - Tópicos Avançados em Algoritmos

	Redes Neurais (D003) IF730 - Sistemas de Tempo Real				
15:00h s	(D218) IF716 - Especificação de Requisitos e Validação de Sistemas (D003) IF730 - Sistemas de Tempo Real	(B020) IF751 - Processamento de Imagens (D218) IF768 - Teoria dos Grafos			(D218) IF702 - Redes Neurais (D220) IF722 - Tópicos Avançados em Engenharia de Software
16:00h s	(D218) IF716 - Especificação de Requisitos e Validação de Sistemas (D003) IF730 - Sistemas de Tempo Real	(B020) IF751 - Processamento de Imagens (D218) IF768 - Teoria dos Grafos			(D218) IF702 - Redes Neurais (D220) IF722 - Tópicos Avançados em Engenharia de Software

6.2.3. Considerando Todas as Disciplinas sendo Lecionadas no CIn

Tempo gasto para gerar os resultados: 0:0:26hs

6.2.3.1. Parâmetros utilizados

Número de gerações: 1000

Número de indivíduos: 100

Número de gerações com soluções aceitáveis para parar: 10

Penalidade para violação de restrição obrigatória: 1000

Percentual da população que compete no torneio: 0,25

Probabilidade de cruzamento: 0,5

Probabilidade de mutação: 0,01

Intervalo de atualização da MEM: 20

Percentual de indivíduos na criação da MEM: 0,2

Percentual de eventos da MEM inseridos num filho gerado: 0,3

6.2.3.2. Restrições Violadas

Restrições Obrigatórias (0)

Restrições Opcionais (6)

A disciplina Infra-estrutura de Software (EC) (IF677) não está no seu turno preferencial (manhã).

A disciplina Introdução à Programação (EC) (IF669) não está alternando o horário das aulas.

A disciplina Introdução à Programação (EC) (IF669) está com aulas em dias seguidos.

A disciplina Matemática Discreta (IF670) não está no seu turno preferencial (manhã).

A disciplina Seminário em Informática Teórica (IF778) não está alternando o horário das aulas.

A disciplina Seminário em Inteligência Artificial (IF707) não está alternando o horário das aulas.

6.2.3.3. Agendamento

Disciplinas do 1º período

	Segunda	Terça	Quarta	Quinta	Sexta
8:00hs	(D002) MA531 - Álgebra Vetorial e Linear para Computação (EC)	(D005) IF668 - Introdução à Computação (EC)	(D005) IF669 - Introdução à Programação (EC)	(D001) MA026 - Cálculo Diferencial e Integral 1	(D005) IF669 - Introdução à Programação (EC)
9:00hs	(D002) MA531 - Álgebra Vetorial e Linear para Computação (EC)	(D005) IF668 - Introdução à Computação (EC)	(D005) IF669 - Introdução à Programação (EC)	(D001) MA026 - Cálculo Diferencial e Integral 1	(D005) IF669 - Introdução à Programação (EC)
10:00hs	(D001) MA026 - Cálculo Diferencial e Integral 1	(Anfiteatro) IF669 - Introdução à Programação (EC)	(D003) IF670 - Matemática Discreta		(D002) MA531 - Álgebra Vetorial e Linear para Computação (EC)
11:00hs	(D001) MA026 - Cálculo Diferencial e Integral 1	(Anfiteatro) IF669 - Introdução à Programação (EC)	(D003) IF670 - Matemática Discreta		(D002) MA531 - Álgebra Vetorial e Linear para Computação (EC)
12:00hs					
13:00hs					
14:00hs					
15:00hs	(D003) IF670 - Matemática Discreta				
16:00hs	(D003) IF670 - Matemática Discreta				
17:00hs					

Disciplinas do 2º período

	Segunda	Terça	Quarta	Quinta	Sexta
8:00hs					
9:00hs					
10:00hs					
11:00hs					
12:00hs			(D002) MA027 - Cálculo Diferencial e Integral 2		(D001) FI006 - Física Geral 1
13:00hs	(D003) IF675 - Sistemas Digitais (EC)	(D001) IF673 - Lógica (EC)	(D002) MA027 - Cálculo Diferencial e Integral 2	(D001) IF672 - Algoritmos (EC)	(D001) FI006 - Física Geral 1
14:00hs	(D003) IF675 - Sistemas Digitais (EC)	(D001) IF673 - Lógica (EC)	(D002) FI006 - Física Geral 1	(D001) IF672 - Algoritmos (EC)	
15:00hs	(D001) IF672 - Algoritmos (EC)		(D002) FI006 - Física Geral 1		(D001) MA027 - Cálculo Diferencial e Integral 2
16:00hs	(D001) IF672 - Algoritmos (EC)		(D002) IF675 - Sistemas Digitais (EC)	(D001) IF673 - Lógica (EC)	(D001) MA027 - Cálculo Diferencial e Integral 2
17:00hs			(D002) IF675 - Sistemas Digitais (EC)	(D001) IF673 - Lógica (EC)	

Disciplinas do 3º período

	Segunda	Terça	Quarta	Quinta	Sexta
8:00hs	(D004) IF679 - Informática e Sociedade (EC)	(D003) FI007 - Física Geral 2	(D003) IF677 - Infra-estrutura de Software (EC)	(Anfiteatro) IF674 - Infra-estrutura de Hardware (EC)	(D002) MA028 - Cálculo Diferencial e Integral 3
9:00hs	(D004) IF679 - Informática e Sociedade (EC)	(D003) FI007 - Física Geral 2	(D003) IF677 - Infra-estrutura de Software (EC)	(Anfiteatro) IF674 - Infra-estrutura de Hardware (EC)	(D002) MA028 - Cálculo Diferencial e Integral 3
10:00hs	(D003) MA028 - Cálculo Diferencial e Integral 3	(D004) IF674 - Infra-estrutura de Hardware (EC)		(Anfiteatro) IF679 - Informática e Sociedade (EC)	(D005) FI007 - Física Geral 2
11:00hs	(D003) MA028 - Cálculo Diferencial e Integral 3	(D004) IF674 - Infra-estrutura de Hardware (EC)		(Anfiteatro) IF679 - Informática e Sociedade (EC)	(D005) FI007 - Física Geral 2
12:00hs					
13:00hs					
14:00hs					
15:00hs	(D002) IF677 - Infra-estrutura de Software (EC)				
16:00hs	(D002) IF677 - Infra-estrutura de Software (EC)				
17:00hs					

Disciplinas do 4º período

	Segunda	Terça	Quarta	Quinta	Sexta
8:00hs				(D003) FI108 - Física Geral 3	
9:00hs				(D003) FI108 - Física Geral 3	
10:00hs		(D005) FI108 - Física Geral 3			
11:00hs		(D005) FI108 - Física Geral 3			
12:00hs					
13:00hs	(D002) IF685 - Gerenciamento de Dados e Informação (EC)	(D004) ET586 - Probabilidade e Estatística	(D001) IF815 - Métodos Numéricos		(D002) IF817 - Interface Hardware-Software
14:00hs	(D002) IF685 - Gerenciamento de Dados e Informação (EC)	(D004) ET586 - Probabilidade e Estatística	(D001) IF815 - Métodos Numéricos		(D002) IF817 - Interface Hardware-Software
15:00hs	(D004) IF815 - Métodos Numéricos			(D003) ET586 - Probabilidade e Estatística	(D002) IF685 - Gerenciamento de Dados e Informação (EC)
16:00hs	(D004) IF815 - Métodos Numéricos	(D001) IF817 - Interface Hardware-Software		(D003) ET586 - Probabilidade e Estatística	(D002) IF685 - Gerenciamento de Dados e Informação (EC)
17:00hs		(D001) IF817 - Interface Hardware-Software			

Disciplinas do 5º período

	Segunda	Terça	Quarta	Quinta	Sexta
8:00hs	(D003) EL215 - Circuitos Elétricos	(Anfiteatro) ES413 - Sinais e Sistemas	(D002) IF690 - História e Futuro da Computação	(D004) IF686 - Paradigma de Linguagens Computacionais (EC)	(D004) IF678 - Infra-estrutura de Comunicação
9:00hs	(D003) EL215 - Circuitos Elétricos	(Anfiteatro) ES413 - Sinais e Sistemas	(D002) IF690 - História e Futuro da Computação	(D004) IF686 - Paradigma de Linguagens Computacionais (EC)	(D004) IF678 - Infra-estrutura de Comunicação
10:00hs	(D004) IF678 - Infra-estrutura de Comunicação	(D003) IF686 - Paradigma de Linguagens Computacionais (EC)		(D003) EL215 - Circuitos Elétricos	(Anfiteatro) ES413 - Sinais e Sistemas
11:00hs	(D004) IF678 - Infra-estrutura de Comunicação	(D003) IF686 - Paradigma de Linguagens Computacionais (EC)		(D003) EL215 - Circuitos Elétricos	(Anfiteatro) ES413 - Sinais e Sistemas
12:00hs					
13:00hs					
14:00hs					
15:00hs					
16:00hs					
17:00hs					

Disciplinas do 6º período

	Segunda	Terça	Quarta	Quinta	Sexta
8:00hs					
9:00hs					
10:00hs					
11:00hs					
12:00hs		(B020) LE530 - Inglês para Computação	(B020) ES203 - Eletromagnetismo 1		(B020) ES232 - Introdução aos Dispositivos Semicondutores
13:00hs	(D218) IF682 - Engenharia de Software e Sistemas	(B020) LE530 - Inglês para Computação	(B020) ES203 - Eletromagnetismo 1		(B020) ES232 - Introdução aos Dispositivos Semicondutores
14:00hs	(D218) IF682 - Engenharia de Software e Sistemas		(D218) ES232 - Introdução aos Dispositivos Semicondutores	(B020) IF689 - Informática Teórica	
15:00hs	(B020) ES203 - Eletromagnetismo 1		(D218) ES232 - Introdução aos Dispositivos Semicondutores	(B020) IF689 - Informática Teórica	(B020) IF682 - Engenharia de Software e Sistemas
16:00hs	(B020) ES203 - Eletromagnetismo 1	(B020) IF689 - Informática Teórica		(D220) LE530 - Inglês para Computação	(B020) IF682 - Engenharia de Software e Sistemas
17:00hs	(B020) ES203 - Eletromagnetismo 1	(B020) IF689 - Informática Teórica		(D220) LE530 - Inglês para Computação	

Disciplinas do 7º período

	Segunda	Terça	Quarta	Quinta	Sexta
8:00hs	(D220) ES344 - Princípios de Comunicação	(D222) ES414 - Servomecanismo	(D220) IF688 - Teoria e Implementação de Linguagens Computacionais	(B020) ES238 - Eletrônica 1	(D222) IF684 - Sistemas Inteligentes
9:00hs	(D220) ES344 - Princípios de Comunicação	(D222) ES414 - Servomecanismo	(D220) IF688 - Teoria e Implementação de Linguagens Computacionais	(B020) ES238 - Eletrônica 1	(D222) IF684 - Sistemas Inteligentes
10:00hs	(D220) ES344 - Princípios de Comunicação		(D222) IF684 - Sistemas Inteligentes	(D222) ES414 - Servomecanismo	
11:00hs	(D222) IF688 - Teoria e Implementação de Linguagens Computacionais		(D222) IF684 - Sistemas Inteligentes	(D222) ES414 - Servomecanismo	(D222) ES344 - Princípios de Comunicação
12:00hs	(D222) IF688 - Teoria e Implementação de Linguagens Computacionais				(D222) ES344 - Princípios de Comunicação
13:00hs	(B020) ES238 - Eletrônica 1				
14:00hs	(B020) ES238 - Eletrônica 1				
15:00hs					
16:00hs					
17:00hs					

Disciplinas do 8º período

	Segunda	Terça	Quarta	Quinta	Sexta
8:00hs					
9:00hs					
10:00hs					
11:00hs					
12:00hs					
13:00hs	(D001) IF676 - Metodologia e Expressão Técnico-científica			(D218) IF819 - Projeto de Co-desenvolvimento Hardware-software	
14:00hs	(D001) IF676 - Metodologia e Expressão Técnico-científica			(D218) IF819 - Projeto de Co-desenvolvimento Hardware-software	
15:00hs	(D220) IF819 - Projeto de Co-desenvolvimento Hardware-software		(D001) IF676 - Metodologia e Expressão Técnico- científica		
16:00hs	(D220) IF819 - Projeto de Co-desenvolvimento Hardware-software		(D001) IF676 - Metodologia e Expressão Técnico- científica		
17:00hs					

Disciplinas eletivas

	Segunda	Terça	Quarta	Quinta	Sexta
8:00hs	<p>(B020) IF705 - Automação Inteligente</p> <p>(D218) IF814 - Metodologia de Hardware/Software Co-desing</p>	<p>(B020) IF781 - Empreendimentos em Informática</p> <p>(D218) IF717 - Planejamento e Gerenciamento de Projetos</p> <p>(D220) IF775 - Tópicos Avançados em Algoritmos</p>	<p>(B020) IF716 - Especificação de Requisitos e Validação de Sistemas</p>	<p>(D218) IF751 - Processamento de Imagens</p>	<p>(B020) IF696 - Integração de Dados e Warehousing</p>
9:00hs	<p>(B020) IF705 - Automação Inteligente</p> <p>(D218) IF814 - Metodologia de Hardware/Software Co-desing</p>	<p>(B020) IF781 - Empreendimentos em Informática</p> <p>(D218) IF717 - Planejamento e Gerenciamento de Projetos</p> <p>(D220) IF775 - Tópicos Avançados em Algoritmos</p>	<p>(B020) IF716 - Especificação de Requisitos e Validação de Sistemas</p>	<p>(D218) IF751 - Processamento de Imagens</p>	<p>(B020) IF696 - Integração de Dados e Warehousing</p>
10:00hs	<p>(B020) IF716 - Especificação de Requisitos e Validação de Sistemas</p>	<p>(B020) IF781 - Empreendimentos em Informática</p>	<p>(B020) IF705 - Automação Inteligente</p> <p>(D002) IF722 - Tópicos Avançados em Engenharia de Software</p>	<p>(B020) IF814 - Metodologia de Hardware/Software Co-desing</p> <p>(D218) IF717 - Planejamento e Gerenciamento de Projetos</p> <p>(D220) IF738 - Redes de Computadores</p>	
11:00hs	<p>(B020) IF716 - Especificação de</p>	<p>(B020) IF781 - Empreendiment</p>	<p>(B020) IF705 -</p>	<p>(B020) IF814 - Metodologia de</p>	

s	Requisitos e Validação de Sistemas (D220) IF702 - Redes Neurais	os em Informática	Automação Inteligente (D002) IF722 - Tópicos Avançados em Engenharia de Software	Hardware/Software Co-design (D218) IF717 - Planejamento e Gerenciamento de Projetos (D220) IF738 - Redes de Computadores	
12:00hs	(D220) IF702 - Redes Neurais (D002) IF778 - Seminário em Informática Teórica	(D226) IF707 - Seminário em Inteligência Artificial	(D226) IF778 - Seminário em Informática Teórica	(D002) IF707 - Seminário em Inteligência Artificial	
13:00hs	(D220) IF751 - Processamento de Imagens (D222) IF722 - Tópicos Avançados em Engenharia de Software		(D220) IF735 - Tópicos Avançados em Arquitetura de Computadores		(D218) IF700 - Percepção Computacional e Reconhecimento de Padrões (D220) IF768 - Teoria dos Grafos
14:00hs	(D220) IF751 - Processamento de Imagens (D222) IF722 - Tópicos Avançados em Engenharia de Software	(D220) IF730 - Sistemas de Tempo Real	(B020) IF696 - Integração de Dados e Warehousing (D220) IF735 - Tópicos Avançados em Arquitetura de Computadores		(D218) IF700 - Percepção Computacional e Reconhecimento de Padrões (D220) IF768 - Teoria dos Grafos
15:00hs	(D218) IF700 - Percepção Computacional e Reconhecimento de Padrões	(D218) IF738 - Redes de Computadores (D220) IF730 -	(B020) IF696 - Integração de Dados e		(D220) IF702 - Redes Neurais (D218) IF775 - Tópicos

		Sistemas de Tempo Real	Warehousing (D220) IF735 - Tópicos Avançados em Arquitetura de Computadores		Avançados em Algoritmos
16:00hs	(D218) IF700 - Percepção Computacional e Reconhecimento de Padrões	(D218) IF738 - Redes de Computadores (D220) IF730 - Sistemas de Tempo Real	(D218) IF768 - Teoria dos Grafos (D220) IF735 - Tópicos Avançados em Arquitetura de Computadores		(D220) IF702 - Redes Neurais (D218) IF775 - Tópicos Avançados em Algoritmos
17:00hs		(D220) IF730 - Sistemas de Tempo Real	(D218) IF768 - Teoria dos Grafos		

7. Capítulo VII – Considerações Finais e Trabalhos Futuros

Ao total, mais de 90 simulações foram realizadas, onde vários parâmetros foram avaliados. Dos resultados obtidos, pudemos tirar as seguintes conclusões.

a) Número de gerações

O número de gerações não precisa ser muito alto (maior que 5.000), pois a população tende a convergir, no pior caso observado, em 2.000 gerações. Sendo que a maioria dos resultados convergiu para um ótimo antes de 1.000 gerações.

b) Número de indivíduos e geração de filhos

Yang e Jat (2011) propuseram um modelo onde a cada geração, apenas um filho seria gerado. Testes com esse modelo mostraram certa limitação quando o número de restrições consideradas aumentou ao passarmos a considerar todas as restrições existentes. Tais restrições envolvem disciplinas obrigatórias que são lecionadas em outros centros e, portanto, o horário já é pré-fixado por eles, professores que estão ocupados por ensinarem matérias em outros centros ou no curso de Ciências da Computação, salas ocupadas pelas aulas do curso de Ciências da Computação, e disciplinas que ocorrem em conjunto com o curso de Ciências da Computação, tendo assim seu horário pré-fixado pelo outro curso.

Para resolver esse problema e ampliar o espaço de busca, em vez de usarmos o modelo proposto por Yang e Jat (2011), a cada geração geramos 10 filhos, dos quais os 4 melhores entrariam para a população e os 4 piores de toda a população seriam descartados. Após implementarmos este novo modelo, obtivemos melhores resultados.

Outra modificação que fizemos foi no número de indivíduos. Yang e Jat (2011) propuseram uma população de 20 indivíduos, mas os testes mostraram que uma população maior, com 100 indivíduos, tende a gerar melhores resultados.

c) Probabilidade de mutação

Testes com a probabilidade de mutação mostraram que uma probabilidade mais baixa que os valores utilizados por Yang e Jat (2011) tende a gerar resultados melhores. Enquanto estes recomendaram uma probabilidade de mutação de 10%, em nossos testes a utilização de uma probabilidade de mutação entre 1% e 5% gerou melhores resultados.

d) Busca local

Ao utilizarmos apenas as duas primeiras etapas da busca local, denominadas por Yang e Jat (2011) de busca local 1, obtivemos resultados ótimos, ou seja, sem nenhuma violação de restrição obrigatória ou opcional, apenas para casos com poucas restrições. Ao considerarmos todas as restrições existentes, a busca local gerava cerca de 8 a 20 violações

de restrições obrigatórias, fora as opcionais. Com a introdução da terceira etapa da busca local, o sistema violava entre nenhuma e 11 restrições obrigatórias. A não utilização de busca local mostrou resultados com várias violações de restrições obrigatórias (mais de 60), não obstante as variações nas probabilidades de mutação, cruzamento e indivíduos selecionados para o torneio e aos percentuais de eventos da MEM utilizados na geração de um filho

e) Percentual de eventos da MEM utilizados para gerar filhos

Os testes indicaram que a taxa de 10% a 30% mostra melhores resultados do que para as taxas muito elevadas. Disso podemos tirar duas conclusões: em primeiro lugar, se gerarmos filhos utilizando um grande percentual da MEM, tenderemos a gerar filhos muito parecidos com os que já existem, reduzindo a capacidade de busca do AG. Em segundo, o fato de duas alocações terem sido tiradas da MEM, não indica que elas serão perfeitas. As alocações armazenadas na MEM são perfeitas em cada indivíduo que origina a MEM. Porém, uma alocação perfeita de um indivíduo pode não ser perfeita se unida a uma alocação perfeita de outro indivíduo, fazendo assim que a MEM gere indivíduos com mais conflitos do que se utilizarmos um baixo percentual de eventos.

f) Configurações ideais

As configurações que mostraram melhores resultados foram utilizando-se uma população de 100 indivíduos, com uma baixa probabilidade de mutação, entre 1% e 5%, e com uma alta penalidade para as restrições obrigatórias, entre 100 e 1000.

g) Sistema como um todo

Apesar de não ter encontrada uma solução que satisfaça todas as restrições obrigatórias e opcionais, os resultados se mostraram promissores. De fato, os resultados gerados pelo sistema, utilizando-se as configurações recomendadas, satisfizeram todas as restrições obrigatórias e geraram menos violações de restrições opcionais que o horário gerado por especialistas humanos. Além disso, o sistema gerou esses resultados em pouco tempo, sendo que em mais de 90% das configurações testadas o tempo gasto para se gerar os resultados não ultrapassou 15 minutos, e em mais de 70% não ultrapassou 10 minutos, enquanto um especialista leva de alguns dias a algumas semanas para montar um resultado que na maioria das vezes também não satisfaz todas as restrições. Os testes executados com as configurações ideais chegaram a encontrar uma solução sem nenhuma violação de restrição obrigatória em menos de 10 segundos.

7.1. Trabalhos Futuros

Planejamos expandir o sistema desenvolvido para acrescentando-lhe as seguintes funcionalidades:

a) Suporte a todos os cursos de uma universidade

Automatizar a geração de tabelas de horários de não apenas um curso, mas de todos os cursos de uma instituição de ensino. O sistema deveria ser expandido incluindo-se noções de disciplinas que ocorrem em mais de um curso, várias turmas que cursam o mesmo período de um curso, porém em horários diferentes, salas que podem ser utilizadas por um curso, mas não por outro, etc.

b) Gerar tabelas a partir de soluções prévias

Possibilitar ao sistema carregar uma solução gerada ao se gerar uma nova solução. Com isso, o sistema pode partir de soluções similares já conhecidas elaboradas ou manualmente por especialistas humanos ou por ele próprio e usar essa informação como uma heurística de partida para se atingir uma nova solução, dado que embora ao longo dos semestres e dos anos as restrições tendem a mudar, como novas disciplinas sendo lecionadas, novas salas, maiores demandas, novos professores com novos horários. Contudo, grande parte das restrições tende a permanecer muito similar e as mudanças não costumam acontecer de forma abrupta, como mudar todas as disciplinas de um curso em um único semestre. Essa mudança é gradual e se dá paulatinamente ao longo de vários anos. Como de um semestre para outro a maior parte das disciplinas, salas e professores tende a ser igual, apresentando as mesmas restrições, usar uma solução passada bem-sucedida pode apresentar um ganho de qualidade da solução gerada e principalmente um ganho de desempenho, pois o ótimo global encontrado para a solução passada provavelmente estará perto do ótimo global para a nova solução no espaço de busca.

c) Interface web

Com uma interface web, em vez de os professores submeterem os horários que eles não podem ou não gostariam de dar aula à secretaria para depois a secretaria cadastrar no sistema, os próprios professores poderiam informar diretamente ao sistema. Isso incorreria em menos erros e problemas, dado que o responsável por cadastrar todas as restrições dos professores pode eventualmente cometer um erro trocando os horários de um professor com o de outro. Os professores teriam um prazo até o qual poderia modificar suas restrições de horário. Após o prazo, o sistema não aceitaria mais alterações pela interface web e geraria as tabelas de horários.

Outro benefício da interface web seria que cada aluno poderia criar uma conta individual onde ele veria os horários, salas e professores de cada uma de suas matérias. Juntamente a este recurso, os professores poderiam informar caso fossem faltar ou chegar atrasados em alguma matéria, e os alunos cadastrados naquela matéria receberiam o aviso do

cancelamento ou atraso da aula por e-mail, evitando que eles fiquem esperando a chegada do professor quando este na verdade não poderá ir por algum motivo como doença, etc.

d) Aplicativos para smartphones

Com um aplicativo smartphone, os professores teriam mais mobilidade para informar aos alunos um cancelamento ou atraso de uma aula. Caso ele tenha sofrido um acidente, tenha ido levar um filho ao hospital ou esteja preso no trânsito, ele poderia informar do seu smartphone o cancelamento ou atraso da aula. Semelhantemente, os alunos receberiam a informação no aplicativo dos seus smartphones que geraria um alerta sonoro. Ao checar o smartphone o aluno ficaria informado sobre a situação da aula.

Referências Bibliográficas

1. Shengxiang Yang e Sadaf Naseem Jat, *Genetic Algorithms With Guided and Local Search Strategies for University Course Timetabling*, 2011
2. O. Rossi-Doria, M. Sampels, M. Birattari, M. Chiarandini, M. Dorigo, L. Gambardella, J. Knowles, M. Manfrin, M. Mastrolilli, B. Paechter, L. Paquete, and T. Stutzle, *A Comparison of the Performance of Different Metaheuristics on the Timetabling Problem*, 2003
3. Rhydian Lewis, *A Survey of Metaheuristic-based Techniques for University Timetabling Problems*, 2008
4. C.Y. Cheong, K.C. Tan, B. Veeravalli, *A Multi-objective Evolutionary Algorithm for Examination Timetabling*, 2009
5. Arindam Chaudhuri e Kajal De, *Fuzzy Genetic Heuristic for University Course Timetable Problem*, 2010
6. Robertus J. Willemen, *School Timetable Construction: Algorithms and Complexity*, 2002
7. D.S. Johnson, C.R. Aragon , L.A. McGeoch, C. Schevon, *Optimization by Simulated Annealing: an Experimental Evaluation I: Graph Partitioning*, 1989
8. M. Dorigo, *Ant Colony Optimization*, <http://www.aco-metaheuristic.org/>