

Universidade Federal do Pampa

Diego Sater da Luz Teixeira

Geração Automática de grade horária usando Simulated Annealing

Alegrete

2015

Diego Sater da Luz Teixeira

Geração Automática de grade horária usando Simulated Annealing

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Ciência da Computação da Universidade Federal do Pampa como requisito parcial para a obtenção do título de Bacharel em Ciência da Computação.

Orientador: Prof. Dr. Fabio Natanael Kepler

Alegrete

2015

Diego Sater da Luz Teixeira

Geração Automática de grade horária usando Simulated Annealing

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Ciência da Computação da Universidade Federal do Pampa como requisito parcial para a obtenção do título de Bacharel em Ciência da Computação.

Trabalho de Conclusão de Curso defendido e aprovado em 10 de Julho de 2015.

Banca examinadora:



Prof. Dr. Fabio Natanael Kepler
Orientador



Prof. Dr. Alessandro Gonçalves
Girardi
Unipampa



Prof. Dr. José Carlos Bins Filho
Unipampa

Resumo

Todo início de período letivo, gestores de instituições de ensino se deparam com um típico problema: montar as grades horárias das turmas, segundo as demandas de aulas de suas disciplinas e considerando as restrições de disponibilidade horária de todos os envolvidos. Conhecido na literatura como *School Timetabling Problem* (STP), este típico problema de otimização combinatória é reconhecidamente complexo por conta do seu elevado número de variáveis e restrições. Tendo em vista atender a demanda por ferramentas de automação de tal problema, neste trabalho desenvolveu-se uma aplicação web geradora de grade horária. Ao efetuar-se experimentos em um modelo de grade horária genérica de uma escola de ensino médio, utilizou-se o algoritmo *Simulated Annealing*(SA) já reconhecido como uma boa alternativa para automatizar a tarefa, o mesmo quando utilizado na aplicação web já mencionada demonstrou algumas limitações para obter ótimos globais, mas de forma geral obteve resultados satisfatórios para ambiente web.

Palavras-chave: Grade horária, *Simulated Annealing*, Aplicação web.

Abstract

Every beginning of school year, educational institutions managers are faced with a typical problem: Elaborate the operating hours of classes, according to the demands of classes in their disciplines and considering the constraints of time availability of all involved. Recognized in the literature as *School Timetabling Problem* (STP), this typical combinatorial optimization problem is admittedly complex because of its high number of variables and constraints. In order to fill the demand for automation tools to such problem, this paper has developed a timetable generator web application. When making up experiments in a general timetable model of a high school, we used the *Simulated Annealing* (SA) algorithm already recognized as a good alternative to automate the task, even when applied in the web application aforementioned showed some limitations to obtain global optima, however got satisfactory results in a web environment.

Key-words: *School Timetabling Problem, Simulated Annealing, Web Application.*

Lista de ilustrações

Figura 1 – Representação de um período (coluna em coloração mais escura) na grade horária (GH)	18
Figura 2 – Representação de uma aula na grade horária (GH)	19
Figura 3 – Grade curricular para uma turma do ensino médio	19
Figura 4 – Exemplo de grade horária (GH) de escola de ensino médio	20
Figura 5 – Timeline da pesquisa com <i>Timetabling</i> (Baseada em figura semelhante do trabalho de Poulsen (2012))	25
Figura 6 – Modelagem dos modelos Rails na aplicação	35
Figura 7 – Algoritmo gerador de grade da aplicação	36
Figura 8 – Grade horária com conflitos em destaque.	38
Figura 9 – Convergência dos custos das 10 grades geradas pela aplicação no experimento 1. Instância de 10 turmas. No eixo x, número ordinal da iteração onde houve atualização do custo da melhor grade até o momento, no eixo y o valor do custo obtido naquela iteração.	39
Figura 10 – Convergência dos custos de 5 grades geradas pela aplicação no experimento 2. Instância de 10 turmas. No eixo x, número ordinal da iteração onde houve atualização do custo da melhor grade até o momento, no eixo y o valor do custo obtido naquela iteração.	41
Figura 11 – Convergência dos custos das 10 grades geradas pela aplicação no experimento 3. Instância de 5 turmas. No eixo x, número ordinal da iteração onde houve atualização do custo da melhor grade até o momento, no eixo y o valor do custo obtido naquela iteração.	42

Lista de tabelas

Tabela 1 – Modelo de Slot	18
Tabela 2 – Configurações da grade para os experimentos realizados	37
Tabela 3 – Configurações do SA para os experimentos realizados	37
Tabela 4 – Ligações dos professor a suas respectivas disciplina e estas ao fator correspondente. <i>Slots</i>	38
Tabela 5 – Resultados para configuração do SA: Temperatura inicial 1000, total de iterações 1000000, experimento 1	39
Tabela 6 – Resultados para configuração do SA: Temperatura inicial 100e20, total de iterações 10000000, experimento 1	40
Tabela 7 – Resultados para configuração do SA: Temperatura inicial 100e5, total de iterações 4000000, experimento 3	41
Tabela 8 – Resultados de cada instância do STP para todos os experimentos . . .	42
Tabela 9 – Grade gerada para configuração do experimento 1 - custo da solução:2	52
Tabela 10 – Grade gerada para configuração do experimento 1 - custo da solução: 2	52
Tabela 11 – Grade gerada para configuração do experimento 1 - custo da solução: 2	53
Tabela 12 – Grade gerada para configuração do experimento 1 - custo da solução: 3	53
Tabela 13 – Grade gerada para configuração do experimento 1 - custo da solução: 3	53
Tabela 14 – Grade gerada para configuração do experimento 1 - custo da solução: 0	54
Tabela 15 – Grade gerada para configuração do experimento 1 - custo da solução: 2	54
Tabela 16 – Grade gerada para configuração do experimento 1 - custo da solução: 2	54
Tabela 17 – Grade gerada para configuração do experimento 2 - custo da solução: 10	55
Tabela 18 – Grade gerada para configuração do experimento 2 - custo da solução: 9	56
Tabela 19 – Grade gerada para configuração do experimento 2 - custo da solução: 9	56
Tabela 20 – Grade gerada para configuração do experimento 3 - custo da solução: 6	57
Tabela 21 – Grade gerada para configuração do experimento 3 - custo da solução: 8	58
Tabela 22 – Grade gerada para configuração do experimento 3 - custo da solução: 4	58
Tabela 23 – Grade gerada para configuração do experimento 3	59

Sumário

1	INTRODUÇÃO	13
2	GERAÇÃO AUTOMÁTICA DE GRADES HORÁRIAS	15
2.1	O School Timetabling Problem (STP)	16
2.2	Conceitos e terminologias	17
2.3	Formulação matemática	20
2.4	Complexidade	21
3	TRABALHOS RELACIONADOS	23
4	SIMULATED ANNEALING	29
4.1	Boltzmann Annealing	30
5	APLICAÇÃO DESENVOLVIDA	33
5.1	Ferramentas utilizadas	34
5.2	Algoritmo da aplicação geradora da GH	35
6	EXPERIMENTOS	37
6.1	Experimento 1	39
6.2	Experimento 2	40
6.3	Experimento 3	41
6.4	Análise dos resultados	42
7	CONCLUSÃO	45
	Referências	47
	APÊNDICE A – APÊNDICE A	51

1 Introdução

Todo início de período letivo, gestores de instituições de ensino - de todos os portes, tipos e níveis - se deparam com um típico problema: montar as grades horárias das turmas, o que pressupõe alocar professores, salas de aulas, laboratórios e quadras esportivas para as turmas de alunos, segundo as demandas de carga horária de suas disciplinas e considerando as restrições de disponibilidade de cada recurso.

Este problema é conhecido na literatura como *School Timetabling Problem*(STP). De acordo com (COOPER T.B.; KINGSTON, 1993), o problema básico do STP consiste em atribuir horários, professores, alunos e salas de aula para uma coleção de turmas, de tal forma que nenhum participante é obrigado a comparecer a duas aulas simultaneamente. Casos reais podem ser muito grandes, com centenas de participantes e centenas de aulas em uma semana com 40 horários disponíveis e restrições variadas.

Segundo (BARDADYM, 1996), a solução manual de um problema de *Timetabling* é uma tarefa árdua e normalmente requer vários dias de trabalho. Essa tarefa árdua acaba se transformando num grande quebra-cabeça, num pesado processo de tentativa e erro que, na primeira combinação encontrada, dá-se por solucionado, mesmo que a qualidade da solução seja péssima. Uma solução péssima implica em alunos e professores prejudicados: uma distribuição de carga horária dissonante dos interesses pedagógicos e maçante para professores e alunos, aulas em dias ou horários indesejados, dentre outras tantas situações que geram insatisfação.

A principal dificuldade, apresentada pelo STP, está no fato de sua natureza combinatória, analisando sob a ótica de um problema de otimização, apesar de já ter sido bastante explorado, continua sendo um desafio (MOURA A.V.; SCARAFICCI, 2004).

Neste trabalho implementou-se uma aplicação web geradora de grade horária (GH) para uma versão simplificada do STP, utilizando o algoritmo *Simulated Annealing*(SA) como estratégia de busca. Foram executados conjuntos de experimentos utilizando tal aplicação, objetivando avaliar as GH obtidas de forma automatizada e o comportamento do algoritmo SA durante estes experimentos.

No Capítulo 2 aborda-se a geração da grade horária, a definição do problema, conceitos, terminologias, complexidade e detalhes relacionados ao *School Timetabling Problem* (STP), no Capítulo 3 faz-se uma revisão dos trabalhos relacionados ao tema deste trabalho, no Capítulo 4, analisa-se o algoritmo *Simulated Annealing* (SA), sua utilização em trabalhos e competições de timetabling, no Capítulo 5, explica-se sobre a aplicação desenvolvida, suas características e requisitos, no Capítulo 6, apresentam-se os experimentos efetuados e os resultados obtidos.

2 Geração automática de grades horárias

Há muitos anos pesquisadores e profissionais de diversas áreas têm dedicado especial atenção a problemas de otimização, com o intuito de obter o melhor aproveitamento possível de recursos, pessoais e materiais, objetivando gerenciar, alocar e distribuir da forma mais racional possível atividades que demandem tais recursos dentro de suas empresas, universidades, laboratórios entre outros.

A mesma motivação é compartilhada por aqueles que estão encarregados de desempenhar as funções acima mencionadas dentro de instituições de ensino (IE), encarregados de construir o melhor arranjo de aulas, considerando as demandas de carga horária, disciplinas, espaços físicos e a disponibilidade de professores dentro de um período de tempo destinado a atividades educacionais.(MOURA A.V.; SCARAFICCI, 2004)

Uma grade horária segundo (POULSEN, 2012) dentro do conceito do STP trata com três entidades: *Atividades, Recursos e Tarefas*.

A grade horária constitui-se inteiramente destas três entidades uma vez que orienta o fluxo do trabalho, definindo onde, como, e quando o profissional do ensino irá desempenhar suas tarefas, qual será a rotina de aulas seguida pelas turmas durante um longo período de tempo, além de servir de “mapa” a orientadores educacionais e gestores da instituição de ensino (IE).

Os elaboradores de grade horária (GH) encarregados de executarem tal tarefa deparam-se com o problema conhecido na literatura como *School Timetabling Problem* (STP). De acordo com (COOPER T.B.; KINGSTON, 1996), o problema básico do STP consiste em atribuir horários, professores, alunos e salas de aula para uma coleção de turmas, de tal forma que nenhum participante é obrigado a comparecer a duas aulas simultaneamente.

Em circunstâncias reais os volume de dados, constantes em uma grade completa, podem ser de um tamanho considerável, chegando a centenas de participantes em centenas de aulas para uma semana com 40 horários disponíveis sob restrições variadas.

Segundo (BARDADYM, 1996), a solução manual de um problema de timetabling é uma tarefa árdua e normalmente requer vários dias de trabalho. Essa tarefa árdua acaba se transformando num grande quebra-cabeça, num pesado processo de tentativa e erro que, na primeira combinação encontrada, dá-se por solucionado, mesmo que a qualidade da solução seja péssima.

Apesar de inexistirem dados oficiais, estima-se que a maioria das instituições de ensino brasileiras não dispõe de uma ferramenta que auxilie neste processo. Desta forma,

dias e até semanas são despendidos nesta tarefa, demandando tempo dos elaboradores, na sua maioria profissionais de pedagogia sem familiaridade com o problema.

2.1 O School Timetabling Problem (STP)

Gotlieb (1962) propôs pela primeira vez uma formulação completa para o STP declarando que o problema consistia em fixar um conjunto de aulas de uma turma num determinado período de tempo, exigindo-se o comparecimento de um único professor. Pertence a uma classe ampla de problemas conhecidos simplesmente por *timetabling*. Esta versão clássica do STP, mostrou-se *NP-complete* por (EVEN S.; ITAI, 1976). Para (ELMOHAMED M., 1997), o problema *timetabling* em si é um problema combinatório de otimização multidimensional, não-euclidiano, multirrestrito e, conseqüentemente, muito difícil de ser resolvido. Alvarez-Valdez R.; Crespo (2002) apontam que a grande quantidade de entidades envolvidas nesse processo (turmas, professores, salas, etc.) resulta num enorme número de variáveis e restrições.

No caso do STP entende-se por *variável* todo item pertencente a um conjunto específico (professores, disciplinas, salas de aula) ou itens associados proveniente destes conjuntos, que possam ser atribuídos a um determinado período de tempo dentro da GH, assim como *restrição* denota uma regra ou definição que delimite as possibilidades de atribuição de determinada variável a um período específico.

De fato a definição do STP e suas características podem variar de país para país, pois cada sistema educacional tem suas características e regras próprias. Isso fica evidente ao se constatar a quantidade de trabalhos que apresentam modelos desenvolvidos especialmente para seus países: Grécia (BELIGIANIS G.N. ; MOUSCOPOULOS, 2008), Espanha (ALVAREZ-VALDEZ R.; CRESPO, 2002), Itália (COLORNI A.; DORIGO, 1998), Austrália (ABRAMSON D.; KRISHNAMOORTHY, 1999), Brasil (OCHI L. S.; SOUZA, 2004) entre outros.

Para (POULSEN, 2012) ao se propor um modelo de STP é fundamental que se relacione os conjuntos envolvidos, bem como as restrições. Segundo (OCHI L. S.; SOUZA, 2004), as restrições, independentemente do seu tipo, são bastante dependentes da instituição de ensino e do sistema educacional de cada localidade. Os autores de forma geral dividem as restrições em 3(três) classes:

- *Organizacionais*: relativas à instituição de ensino, que tratam da gestão de recursos, bem como do atendimento à legislação presente, como:

Carga horária máxima semanal de professores: por questões de ordem legal, professores podem ter sua carga horária semanal limitada.

Número de janelas (intervalo em que algum agente da GH se encontra ocioso): a quantidade de janelas para cada professor, por exemplo, impacta salário a pagar.

- *Pedagógicas:* demandas importantes para o bom aproveitamento das aulas como:

Atendimento de N-uplas (aulas alocadas em períodos contíguos na GH): em alguns casos, atender ao pedido de aulas N-uplas pode representar um melhor rendimento pedagógico da turma, considerando-se para tais caso ser importante duas ou mais aulas seguidas de uma mesma matéria.

Limitação de aulas diárias: disciplinas de grande carga horária podem demandar um limite diário de aulas com o objetivo de não deixar a aula muito cansativa.

Preferência de professores: por razões pedagógicas, a *Instituição de Ensino* (IE) pode preferir um determinado professor para lecionar determinada disciplina em detrimento de outro.

- *Pessoais:* solicitadas de acordo com a preferência e necessidade dos professores, como:

Indisponibilidade em determinados dias: o professor pode estar indisponível para lecionar em determinados dias.

Preferência de dias: o professor, mesmo tendo disponibilidade, pode preferir lecionar em determinado dia em detrimento de outro.

2.2 Conceitos e terminologias

Ao longo deste trabalho, conceitos e terminologias são empregados, determinando a necessidade de explaná-los devidamente para melhor compreensão do contexto no qual estão inseridos. Buscou-se utilizar estes termos pelo fato de serem consagrados pela literatura ou pelo uso comum. Em uma instituição de ensino, cada aluno cursa uma série, ano ou etapa de um determinado curso. Cada série, ano ou etapa representa um determinado nível do processo contínuo de ensino-aprendizagem e dispõe de um rol de disciplinas que formam a *grade curricular*. Numa grade curricular, cada disciplina tem um número de aulas semanais. A fim de organizar os alunos que cursam cada série, a instituição de ensino agrupa os alunos através de turmas. Uma série, ano ou etapa pode ser ofertada por meio de uma ou mais turmas. A quantidade de turmas é determinada por vários fatores: demanda de alunos por determinados turnos, capacidade física da IE, limitações de ordem pedagógica. Cada disciplina é oferecida aos alunos de uma turma de acordo com a exigência de carga horária expressa na grade curricular da etapa a qual estão vinculados. Cada disciplina é ministrada por um único professor numa determinada turma.

Slot: O termo *slot* designa a associação de uma professor a uma única disciplina ilustrado na [Tabela 1](#), convenção amplamente utilizada em pesquisas de *Timetabling*, para fins de implementação contém os *ids* dos agentes os quais referencia.

Tabela 1 – Modelo de Slot - Valores referentes aos *ids* de professor e disciplina extraídos da aplicação desenvolvida.

Id	Professor	Disciplina
86	11	7

Período de aula: É a janela de tempo em que uma aula é ministrada, independente do dia da semana. O período de aula tem uma hora de início, uma hora de fim e está dentro de um turno (manhã, tarde ou noite). Na [Figura 1](#) podemos ver em destaque (cor mais escura) um exemplo de período.

Figura 1 – Representação de um período (coluna em coloração mais escura) na grade horária (GH)

	Dias da Semana																								
	SEG					TER					QUA					QUI					SEX				
	Períodos																								
Turmas	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
101																									
102																									
103																									
104																									
105																									
106																									
107																									
108																									
109																									
110																									

Aula: Consiste em uma célula da grade horária, isto é, a intersecção entre um dia da semana e um período de aula em que uma disciplina é ministrada por um professor para uma turma. Na [Figura 2](#), uma intersecção de dia da semana e um período aula é ilustrada dentro do contexto de uma grade completa.

Turma: É um agrupamento de alunos que cursam uma grade curricular conjuntamente ao longo de um período letivo, cumprindo uma grade horária. Exemplo: Turma 102, Turma 103,...

Cada turma possui sua própria grade com componentes curriculares adequados ao seu curso, no entanto para fins de geração da grade horária automática pela aplicação desenvolvida neste trabalho considerou-se uma instância onde todas as turmas compartilham a mesma grade curricular. Para os experimentos efetuados, considerou-se um conjunto de professores que compartilha o mesmo conjunto de turmas e não há conflitos de espaços físicos.

Figura 2 – Representação de uma aula na grade horária (GH)

Turmas	Dias da Semana																								
	SEG					TER					QUA					QUI					SEX				
	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
101																									
102																									
103																									
104																									
105																									
106																									
107																									
108																									
109																									
110																									

Grade curricular : Também chamada de matriz curricular, é a relação de disciplinas com suas respectivas cargas horárias de uma determinada série, ano ou etapa. Na [Figura 3](#) é apresentada a grade curricular de uma das turmas de uma instituição de ensino.

Figura 3 – Grade curricular para uma turma do ensino médio

Grade curricular	
Disciplinas	fator
matemática	5
portugues	4
física	3
química	2
biologia	2
história	2
geografia	1
inglês	2
ed. física	2
informática	2
carga horária total	25

Cada disciplina que compõe a grade curricular possui um número inteiro associado chamado *fator de repetição*, este define a quantos períodos será atribuída tal disciplina, dentro de uma grade horária válida para uma turma específica.

Grade horária: É uma matriz em que linhas são as turmas e as colunas maiores são os dias da semana, subdivididas em períodos de aula. É, portanto, onde os horários de turmas, professores e disciplinas são alocados. Uma GH repete-se semanalmente ao longo do período letivo, um exemplo pode ser visto na [Figura 4](#), onde cada célula correspondente a uma turma em determinado período, recebe a referência para um dado *slot* contendo a ligação entre um professor e uma disciplina.

Figura 4 – Exemplo de grade horária (GH) de escola de ensino médio

Turmas	Dias da Semana																								
	SEG					TER					QUA					QUI					SEX				
	Períodos																								
	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
101																									
102																									
103																									
104																									
105																									
106																									
107																									
108																									
109																									

2.3 Formulação matemática

Uma formulação matemática a seguir, expressa um modelo básico para o STP:

- T : conjunto de turmas de alunos que participam da formação da grade horária;
- D : conjunto das disciplinas da grade curricular;
- P : conjunto dos professores que lecionarão as aulas das disciplinas do conjunto D ;
- S_{dp} : conjuntos de slots (ligações de uma disciplinas aos professores que a lecionam);
- D_t : conjunto de disciplinas $d \in D$ da turma $t \in T$;
- D_p : conjunto de disciplinas $d \in D$ do professor $p \in P$;
- S : conjunto de dias da semana;
- P : conjunto de períodos de aula;
- $AC_{|T| \times |P| \times |S|} \in \{0,1\}$: matriz da grade horária das turmas, onde $AC_{tps} = 1$ indica se a turma $t \in T$ deve ter aula no dia $s \in S$ e período $p \in P$, $AC_{tps} = 0$ caso contrário;
- $M_{|D_d|}$: vetor de aulas semanais das disciplinas das turmas $t \in T$, onde M_{D_d} indica o número de aulas semanais da disciplina $d \in D$, conforme grade curricular da turma $t \in T$;
- $Smax_{|D_t|}$: vetor de número máximo de aulas diárias de uma disciplina, onde $Smax$, indica o número máximo de aulas diárias aceitável para uma disciplina $t \in D_t$ de uma turma $t \in T$.

$P_{dt} \in \{0,1\}$: 1 se o professor $p \in P$ possui um slot com a disciplina $d \in D$ integrante da grade curricular da turma $t \in T$, 0 caso contrário.

$$X_{sdp} = \begin{cases} 1, & \text{se a disciplina } s \in S; \text{ é ministrada no dia da semana } d \in D; \text{ e período de aula } p \in P; \\ 0, & \text{caso contrário.} \end{cases}$$

O modelo do STP abordado neste trabalho possui a seguinte função:

$$\text{minimizar } \mathbf{Z} = \sum_{t \in \mathbf{T}} \sum_{s \in \mathbf{S}} \mathbf{P}_{st}$$

2.4 Complexidade

Segundo (BROOKSHEAR, 2003), a complexidade em um problema é diretamente determinada pelas características do algoritmo que o resolve, neste sentido a menor complexidade de um problema será a própria complexidade do algoritmo mais simples que o soluciona. Ainda segundo o autor, dentre diversas maneiras de definir complexidade, a forma mais precisa para avaliação será obtida quando mensurada do ponto de vista do esforço computacional, ou seja, pelo tempo necessário para execução eficaz de um algoritmo aplicado a resolução de um problema.

Assim o tempo dedicado a uma resolução poderá ser definida em termos de uma função $O(f(n))$, sendo este um limite para o qual nenhum outro algoritmo seja capaz de resolver um problema de n variáveis, em um tempo menor.

Conforme definições apresentadas por (GOODRICH, 2004), a classe de complexidade chamada de polinomial (P) é o conjunto de todos os problemas de decisão, que podem ser limitados por uma função polinomial em seu pior caso, ou seja, existe um algoritmo que encontra convergência em tempo $P(n)$, onde n é o número de variáveis do modelo e $P(n)$ o polinômio da função. De forma similar define-se um algoritmo como eficiente, quando o tempo de execução for da ordem $O(nk)$ onde k é uma constante maior que zero ($k > 0$).

Uma segunda classe de problemas são chamados de NP (não polinomiais), para o qual não existe uma função $O(f(n))$, que possa limitar no seu pior caso, o tempo de resolução qualquer que seja o método. De forma geral em um problema NP, o tempo de resolução cresce exponencialmente o seu número de variáveis.

Para os casos de valor elevado de n , o problema dificilmente será resolvido utilizando métodos exatos, com esforço computacional aceitável. Além das funções polinomiais (P) e não polinomiais (NP), são comuns identificar problemas em tempos $O(n^2)$ onde a complexidade é proporcional ao quadrado do número das variáveis, ou $O(n \log n)$ n vezes o logaritmo das variáveis do problema.

Segundo (PILLAY, 2013) a categoria do problema em estudo é uma classe NP de problemas.

Para contornar o problema do alto custo computacional dos métodos exatos, que podem tornar impossível chegar a uma solução em tempo aceitável, são utilizados os métodos heurísticos, estes capazes de encontrar soluções ótimas ou muito próximas do ótimo em muitos casos. (MONTEVECHI J. A. B. ; BATEMAN, 2002) apresentam tais métodos como meio eficiente de desenvolver soluções aproximadas, com baixo custo computacional, sendo ainda vantajosos pela simplicidade de implementação e modelagem flexível, para adequar-se aos mais diversos gêneros de variáveis. Conhecer o histórico de trabalhos relacionados constitui, portanto um passo importante do processo de conhecimento do STP.

3 Trabalhos relacionados

Conforme mencionado em (PILLAY, 2013) toda pesquisa recente visando solucionar o School Timetabling Problem (STP) está focada em métodos heurísticos que simulam a solução manual (ANDREA, 1999). Estratégia aplicada na prática por (GANS, 1981). No entanto desde que foi proposto, várias estratégias já foram consideradas. A medida que a área de pesquisa do STP foi sendo desenvolvida novos métodos foram sendo mais frequentemente adotados e novos algoritmos foram derivando destes primeiros em busca da melhor opção, destacando-se os seguintes:

Metodologias que se utilizam de *teoria dos grafos* foram apresentadas nos trabalhos de (WERRA, 1971) e (COOPER T.B.; KINGSTON, 1993).

GRASP (Greedy Randomized Search Procedure) foi aplicado em trabalhos objetivando obter resultados sem preocupação com o tempo de execução tal como o de Moura A.V.; Scaraficci (2004).

Satisfação de restrições (SAT) foram aplicadas por (AVELLA P.; D'AURIA, 2007), assim como (ABRAMSON D.; KRISHNAMOORTHY, 1999) utilizou um algoritmo genético, utilizando paralelismo para melhorar o tempo de execução do processo.

Em alguns casos abordagens híbridas foram adotadas combinando duas ou mais metodologias em uma única implementação (OCHI L. S.; SOUZA, 2004).

Em outros casos devido ao custo computacional envolvido durante a execução de tais aplicações, métodos distribuídos também têm sido investigados.

No caso de (COLORNI A.; DORIGO, 1998) uma estratégia em três fases distintas, cada uma cuidando de um tipo de restrição foi utilizada.

Como mencionou (SANTOS H.G.; OCHI, 2012) o uso de métodos exatos para resolver instâncias reais do STP demanda quantidades de tempo muito superiores ao aceitável, e conforme comprovou-se pelos trabalhos de (BELIGIANIS G.N. ; MOUSCOPOULOS, 2008) e (BURKE E. K.; PETROVIC, 2002) o uso de métodos heurísticos, por outro lado, tem se mostrado uma excelente alternativa prática.

No entanto, para se ter um parecer definitivo sobre as saídas de um algoritmo heurístico, os resultados obtidos com tais algoritmos para instâncias do STP ainda dependem fortemente de uma avaliação empírica. Em muitos trabalhos, a qualidade da grade horária proposta é medida pela sua aceitação pela escola, que é muito subjetiva. Idealmente, cada nova heurística deve ser comparada com métodos anteriores propostos aplicados a instâncias de *benchmark*.

Na [Figura 5](#) pode-se ter uma visão geral de mais de meio século de pesquisa destacando-se, características a serem tratadas, dimensões, estratégia adotada e o tempo que cada caso levou para obter a saída. O presente trabalho foi incluído na figura com o propósito de situá-lo em um contexto mais amplo dentro da pesquisa realizada até o momento na área.

Para o caso específico do STP o que ocorre na realidade é que cada grupo de pesquisadores/autores consideram variantes diferentes do problema (por vezes, apenas ligeiramente diferentes), o que dificulta possíveis comparações. Para minimizar este problema, um esforço (POST G.;AHMADI, 2012) está sendo feito para padronizar e publicar conjuntos de dados de referência na WEB (SANTOS H.G.;OCHI, 2012).

Outra iniciativa de forte impacto para a área realizada desde 2002, dentro do projeto *Metaheuristics Network*, patrocinado pela *Improve Human Potential* é uma competição conhecida como *International Timetabling Competition* (METAHEURISTICS...), focada inteiramente em métodos de produção automatizada de quadros de horários. Este evento foi um importante marco para a pesquisa nesta área, pois tradicionalmente, poucos autores realizavam experimentos computacionais comparando diferentes métodos de solução (OCHI L. S.; SOUZA, 2004), os resultados demonstrados em tais competições podem ser considerados o estado da arte em termos de aplicações práticas para solução de todos os tipos de problemas de *Timetabling*, incluindo o *School Timetabling Problem* (STP).

Como estas competições demonstraram, os algoritmos heurísticos consistem do estado da arte para solução do *School Timetabling Problem* (STP), pode-se destacar dentre várias estratégias utilizadas o algoritmo conhecido como *Simulated Annealing* (SA), escolha mais recorrente entre os vencedores das competições já utilizado por três campeões de tais competições internacionais (ITCs).

A primeira das competições ocorreu em 2003 e foi vencida por Kostuch, com uma abordagem baseada em *Simulated Annealing* (SA) de 3 fases: geração, busca e melhoria. A segunda competição ocorreu em 2007 e foi composta por três linhas de trabalho diferentes uma para cada caso de timetabling, entre eles o STP, vencidas em sua maioria por Müller, novamente com uma abordagem baseada no algoritmo SA. A última aconteceu em 2012 e foi vencida por Abner usando-se a abordagem conhecida como busca local interativa do SA.

Atentando-se ao fato acima exposto da inexistência de uma padronização rígida, nem formulações que indiquem quais variáveis de decisão são obrigatoriamente adotadas, buscou-se neste trabalho modelar o STP de forma simplificada considerando características de trabalhos semelhantes que já tenham obtido resultados utilizando SA.

O objetivo deste trabalho, que o diferencia dos já mencionados, é verificar o comportamento do algoritmo *Simulated Annealing* (SA) quando utilizado por uma aplicação web no processo de geração automática de grade horária, destacando as seguintes características básicas:

Simplificação do modelo de STP tradicionalmente utilizado, dado o objetivo de verificar os resultados obtidos através de experimentos executados em uma aplicação web

que utilize o algoritmo *Simulated Annealing (SA)*.

Os conflitos de um mesmo professor em um mesmo período em turmas diferentes [Figura 8](#) como critério de penalização de uma solução, em outras palavras o professor é o agente principal na definição de atribuição ou não de certo *slot* a determinado período.

4 Simulated Annealing

Métodos heurísticos, de caráter geral, com capacidade para escapar de ótimos locais, podem ser baseados em *busca local* ou *busca populacional*. Os métodos baseados em busca local são fundamentados na noção de vizinhança: Dada uma solução S , diz-se que S' é um vizinho de S , se S' é obtido de S a partir de um movimento (M), isto é: $S' \leftarrow S \oplus M$.

Os algoritmos que utilizam busca populacional também conhecidos como evolutivos, tem como destaque os algoritmos genéticos ainda bastante usados em problemas de otimização combinatória.

O *annealing* (anelamento ou têmpera) de certos materiais consiste em submetê-los inicialmente a altas temperaturas e reduzi-las gradualmente até atingirem, com aumentos e reduções do estado de energia, o equilíbrio térmico, tornando-os assim, consistentes e rígidos.

A idéia de aplicar este método para resolver problemas de otimização combinatória surgiu bem mais tarde no trabalho de (KIRKPATRICK S. C. D.; GELATT M. P.; VECCHI, 1983).

De forma análoga ao processo físico de anelamento, no qual o resfriamento gradativo de um material a partir de uma alta temperatura inicial, leva o material a estados mínimos de energia. Estes estados são caracterizados por uma perfeição estrutural do material congelado que não se obteria caso o resfriamento não tivesse sido gradativo. A técnica matemática de Simulated Annealing faz uma simulação algorítmica de tal processo. Sob outras condições menos cuidadosas de resfriamento, o material se cristalizaria com uma energia "localmente mínima", apresentando imperfeições estruturais. A esse processo cuidadoso de resfriamento dá-se o nome de *annealing*.

A origem da técnica de otimização conhecida por *Simulated Annealing*(SA) (têmpera simulada ou anelamento simulado) surgiu em 1953, quando foi usada para simular em um computador, o já mencionado processo de *annealing* de cristais.

Conforme mencionado em (SEVERO L.C.;GIRARDI, 2013) o algoritmo SA usa uma *função de resfriamento* $T(t)$, que liga um instante t a uma temperatura T , decrementando T conforme t é incrementado. A cada iteração, novos passos são aleatoriamente dados baseado em uma *função probabilística de geração de estados* $G(x)$, guiando a novos estados dentro do espaço de soluções. Se um passo levar a um estado pior, ele somente será efetivamente tomado, i.e. o novo estado será aceito, com uma probabilidade menor que 1. Estados com melhores soluções sempre são aceitos. Esta probabilidade é dada por

Algorithm 1: Pseudo-código do algoritmo Simulated Annealing

Entrada: TamanhoDoProblema, $iterações_{max}$, $temp_{max}$
Saída: S_{melhor}
inicio
 $S_{corrente} \leftarrow \text{CriaSoluçãoInicial}(\text{TamanhoDoProblema});$
 $S_{melhor} \leftarrow S_{corrente};$
repita
 $S_i \leftarrow \text{CriaSoluçãoVizinha}(S_{corrente});$
 $temp_{corrente} \leftarrow \text{CalculaTemperatura}(i, temp_{max});$
if $\text{Custo}(S_i) \leq \text{Custo}(S_{corrente})$ **then**
 $S_{corrente} \leftarrow S_i;$
if $\text{Custo}(S_i) \leq \text{Custo}(S_{melhor})$ **then**
 $S_{melhor} \leftarrow S_i;$
até $i = iterações_{max};$
 $\exp \frac{\text{Custo}(S_{corrente}) - \text{Custo}(S_t)}{temp_{corrente}} > \text{rand}() S_{corrente} \leftarrow S_i;$
fin
return $S_{melhor};$

uma *função de aceitação* que atende ao seguinte critério:

$$h(\Delta F) = \frac{1}{1 + \exp(\Delta F/T)}$$

Onde, $\Delta F = F_{t+1} - F_t$ representa a variação da função custo calculada em dois passos consecutivos F_{t+1} e F_t

O algoritmo portanto é capaz de alcançar a solução ótima desde que escolha a *função de resfriamento* e a *função probabilística de geração de estado* adequadas. Caso a temperatura na *função de resfriamento* decresça rápido demais, a busca será executada muito rápido, no entanto o algoritmo SA não poderá garantir encontrar a solução ótima global. Este cenário é aceitável no caso de requerer-se um curto espaço de tempo para obter a solução e o espaço de soluções seja bem conhecido ou possui grande dimensionalidade.

4.1 Boltzmann Annealing

O *Boltzmann annealing* é o clássico algoritmo *simulated annealing*, usando princípios físicos para escolher a *função probabilística de geração de estado* de maneira a assegurar a convergência ao mínimo global.

$$T_{Boltz}(t) = \frac{T_0}{\log(t)}$$

onde T_0 é a temperatura inicial, e t é o passo ou iteração atual.

Se $E < 0$, o deslocamento é incorporado ao estado do sistema, que é então utilizado no passo seguinte, caso contrário, isto é, se $\Delta E > 0$, a aceitação ou não do deslocamento passa a ser uma decisão probabilística, segundo a função de Boltzmann: $e^{\Delta/(kT)}$, onde k = constante de Boltzmann e T = temperatura atual.

Os estados possíveis de um metal correspondem a soluções do espaço de busca a energia em cada estado corresponde ao valor da função objetivo energia mínima (se o problema for de minimização ou máxima, se de maximização) corresponde ao valor de uma solução ótima local, possivelmente global.

A estratégia de uso do SA passa, necessariamente, pela correta definição de valores para cada parâmetro. [Johnson \(1998\)](#) sugere algumas diretrizes para a parametrização do SA:

O comprimento da simulação deve ser longo e o aumento pode ser feito aumentando-se a taxa de resfriamento α , e a temperatura inicial T , não é necessário gastar muito tempo utilizando altas temperaturas em que quase todos os movimentos são aceitos, parece ser melhor realizar uma simulação longa (taxa de resfriamento muito próximo de 1, e elevado número de iterações na temperatura T do que realizar pequenas curtas simulações (menor taxa de resfriamento e menor número de iterações). Essa observação considera que os tempos em ambos os casos são compatíveis, é sempre preferível iniciar com uma boa solução que usa algum conhecimento do que iniciar a busca com uma solução gerada aleatoriamente, a parametrização do SA depende não só do problema em questão, mas também do tipo e do tamanho da instância que está sendo considerada.

Segundo ([MICHALEWICZ Z.;FOGEL, 2004](#)), o SA mantém semelhanças com o *stochastic hill climber* (SHC) (subida da montanha estocástica), outro algoritmo utilizado para efetuar busca por soluções ótimas globais em problemas de otimização, diferindo principalmente na mudança de valor do parâmetro T (temperatura), o algoritmo SA executa até que certo estado externo seja alcançado, em contraste com o SHC em que a busca local realizada sempre procura um ponto melhor para S dada sua natureza gulosa.

Ampliando um pouco esta análise pode-se dizer que o algoritmo SA busca apenas um novo ponto na vizinhança de X que satisfaça o requisito da atual temperatura T , a qual é atualizada a cada iteração, assim seu procedimento por vezes aceita soluções não tão boas, buscando sair assim dos ótimos locais.

Aplicações utilizando o algoritmo SA já tem sido desenvolvidas tais como a de ([POULSEN, 2012](#)), estas e a recorrência no uso de tal estratégia de busca nas competições internacionais de timetabling, embasaram a escolha deste algoritmo como principal peça no mecanismo da aplicação desenvolvida neste trabalho.

5 Aplicação desenvolvida

De uma aplicação geradora de grade horária espera-se fundamentalmente: inserção de dados eficiente, tempo de processamento dos dados reduzido, geração de uma grade consistente com o menor número de conflitos possível como saída.

O método de inserção das informações de forma correta em qualquer sistema que proponha-se tratar instâncias do STP consiste de uma questão a ser decidida com cuidado, devido ao volume de dados relativos a todos os agentes e às restrições consideradas para uma instância de uma GH. Já existem iniciativas que visam agilizar tal processo, destacando-se a de (SANTOS H.G.;OCHI, 2012) que propôs uma notação XML específica para o problema, conhecida como *XHSTT (Extensible High School TimeTabling)*, tem se mostrado promissora mas por estar ainda em desenvolvimento, tem sido usada para uma gama ainda muito restrita de instâncias do STP.

Neste trabalho implementou-se um protótipo de uma aplicação web, que possibilite o recebimento de dados relativos a determinada grade curricular: disciplinas, turmas, professores e ligações de uma disciplina com um professor (*Slot*), inseridos através de uma interface simplificada em um navegador, tais dados são armazenados em um banco de dados previamente modelado, possibilitando assim a recuperação dos mesmos para posterior uso do algoritmo que tem como saída uma grade horária com conflitos minimizados.

As restrições para a instância do STP no presente trabalho têm o propósito de possibilitar a geração de uma grade viável, uma GH é assim considerada quando todas as suas aulas forem alocadas de forma que o menor número possível das seguintes restrições, for violada:

1. As aulas semanais de cada disciplina devem ser atendidas;
2. O número máximo de aulas semanais de cada disciplina (fator) não deve ser ultrapassado;
3. Cada turma deve ter aula em todos os períodos da sua grade horária semanal;
4. Um professor pode ter apenas uma aula por período;
5. Uma turma pode ter apenas uma aula por período.
6. Cada professor poderá atender em uma semana de aulas um número máximo de turmas seguindo a seguinte formulação:

T_{maxtp} = Máximo de turmas que o professor poderá atender por semana. T_p = Total de períodos da GH do conjunto de turmas participantes da mesma. F_{ds} = Fator da

disciplina ministrada pelo professor.

$$T_{maxtp} \leq \frac{T_p}{F_{ds}}$$

Uma vez que para determinado professor, este limite de turmas seja alcançado, um novo professor deve ser adicionado para tal disciplina, dividindo-se assim o atendimento de uma mesma disciplina entre dois professores, para um conjunto de turmas em uma mesma GH.

5.1 Ferramentas utilizadas

Para obtenção dos resultados foi utilizado um notebook Acer Aspire E1-571-6601 com as seguintes especificações de hardware: processador modelo Intel® Core™ i3-2348M 2.3GHz, 3MB L3 Cache, 4Gb de memória RAM e 300 Gb de espaço livre em disco.

Para o desenvolvimento da aplicação web onde os experimentos foram efetuados, utilizou-se de um framework web conhecido como *Ruby on Rails (RoR)* (HANSSON D.H. ; THOMAS, 2007), o mesmo utiliza a arquitetura *Model-View-Controller (MVC)* de desenvolvimento, optou-se pelo mesmo em função do mesmo possibilitar o desenvolvimento ágil e ser de grande uso no mercado de desenvolvimento web.

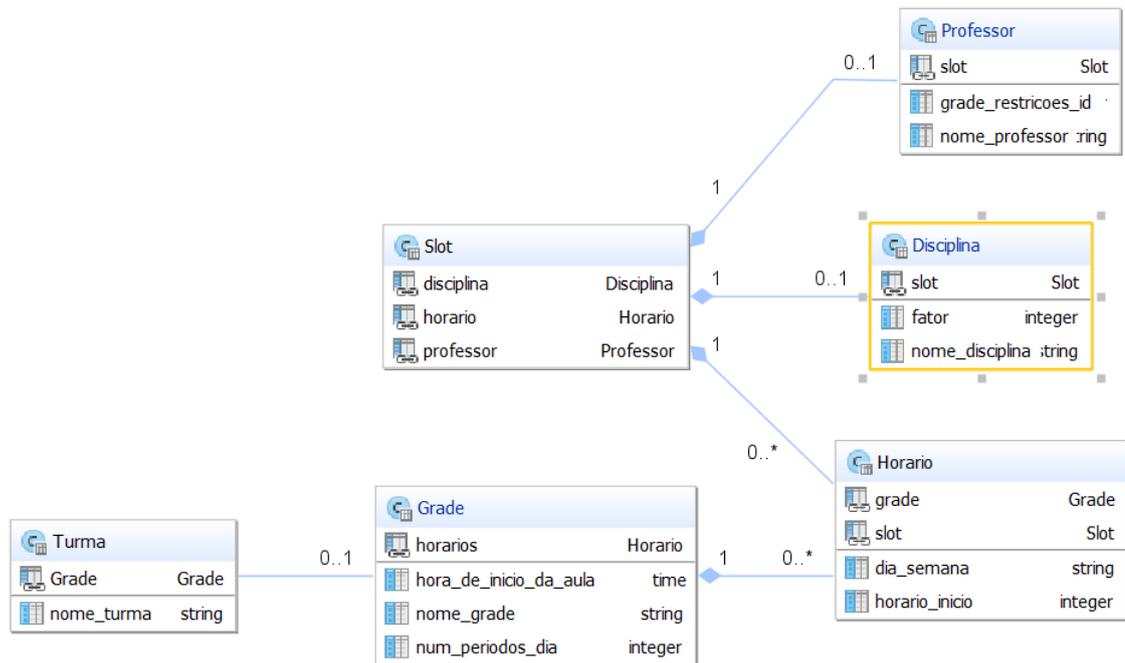
Considerando a estrutura do MVC foi necessária especial atenção aos detalhes de implementação referentes a modelagem de dados que possibilitassem a recuperação eficiente dos mesmo para processamento dentro do algoritmo SA. A modelagem das entidades da aplicação estruturadas conforme são organizadas dentro do RoR pode ser visto na [Figura 6](#). O desenvolvimento de aplicações web, assim como o funcionamento do RoR, dada a utilização do MVC segue a seguinte divisão de tarefas:

Model: Estrutura onde os modelos de dados ou entidades são definidos e regras de acesso e relacionamento entre as mesmas são definidas.

Controller: Conjunto de métodos controlam as operações efetuadas com as entidades definidas no Model, estabelecem o que é chamado de regras de negócio, ou como os dados serão usados e posteriormente enviados à *View* para exibição.

View: Métodos de exibição do resultado do processamento dos dados pelo *Controller*.

Figura 6 – Modelagem dos modelos Rails na aplicação



5.2 Algoritmo da aplicação geradora da GH

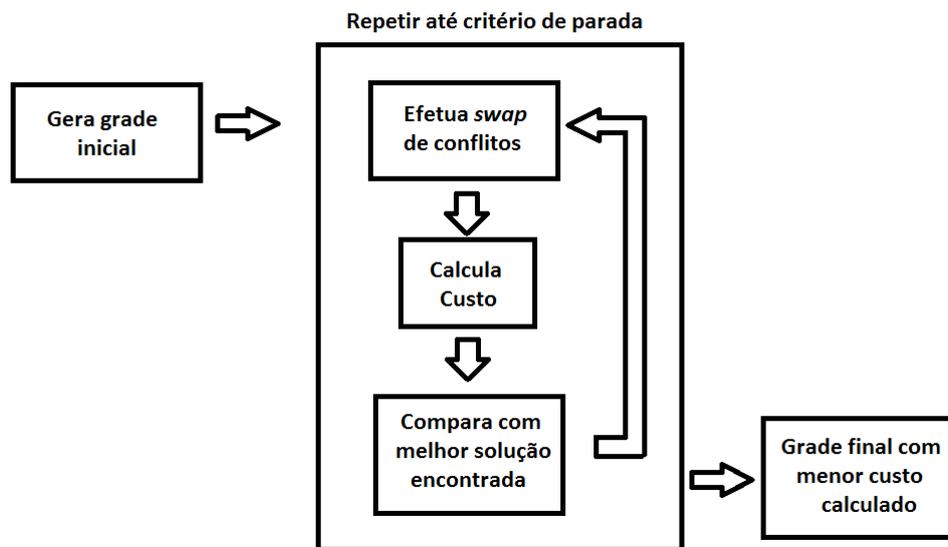
A geração de uma nova solução dentro do espaço de buscas no contexto do STP é conhecida como *swap*, consiste da alteração do “arranjo” de *slots* que possuem conflito entre si em um período específico, para que uma penalidade seja eliminada para a próxima grade gerada.

Conforme ilustrado na [Figura 7](#) o algoritmo de geração da grade automatizada efetua uma sequência bem definida de tarefas. Cada uma delas é constituída de sub-tarefas menores descritas abaixo nos seguintes passos:

- Uma grade inicial é formada organizada de maneira que os *slots* contenham as disciplinas definidas na grade curricular repetidas o número de vezes igual ao fator definido na inserção dos dados, em ordem sequencial.
- O seguinte algoritmo se repete até ser alcançado o número máximo de iterações (critério de parada):
 1. O custo da grade corrente é calculado, e associado a mesma por uma estrutura nos moldes de uma *Hash*.
 2. O mesmo custo é comparado com o da grade de menor custo (melhor até o momento).
 - a) **Se** a grade corrente for melhor (custo mais baixo) que a melhor grade até o momento, a grade corrente assume como melhor grade para futura comparação.

- b) **Senão** é aceita conforme a probabilidade expressa na [Capítulo 4](#).
 3. Efetua um *swap* na ordem dos *slots* em conflito para nova iteração.
 4. Até o critério de parada ser alcançado (número máximo de iterações).
- Grade final é exibida ao usuário.

Figura 7 – Algoritmo gerador de grade da aplicação



Definiu-se como principal indicador de qualidade de uma solução a quantidade de conflitos de atribuição de um mesmo *Slot* mais de uma vez, dentro de um mesmo período em um mesmo dia, cada conflito penaliza em 1 ponto o custo da solução, configurando-se assim o problema como sendo de minimização conforme mencionado na [seção 2.3](#) e utilizado na realização dos experimentos a seguir.

6 Experimentos

Adotou-se neste trabalho uma metodologia de experimentos que visa verificar o comportamento das grades obtidas ao utilizar o algoritmo *Simulated Annealing* (SA) como estratégia de busca. Para fins de análise dos resultados buscou-se variar a parametrização do SA, utilizando diferentes valores de temperatura e número de iterações de forma a melhorar a qualidade da solução, neste caso uma grade horária com menor custo, logo o objetivo é reduzir o número de conflitos de horário [Figura 8](#). Neste trabalho popôs-se 3 instâncias do STP com diferentes características [Tabela 2](#) para 3 parametrizações diferentes do SA como vemos em [Tabela 3](#).

Nas instâncias do STP adotou-se uma estratégia de incremento no número de turmas até o limite que o número de professores “contratados” pela instituição simulada poderia atender, em outras palavras definiu-se um grupo de professores (corpo docente) e por consequência de *Slots* também, posteriormente efetuaram-se experimentos para conjuntos diferentes de turmas para este mesmo corpo docente, sempre levando em consideração a última restrição referida no [Item 6](#) do [Capítulo 5](#).

Os dados cadastrados com a finalidade de efetuar os experimentos objetivaram simular uma GH real, definiu-se portanto turmas com nomes numéricos de 101 a 112, disciplinas de uma grade curricular genérica: Matemática, Português, Física, Química, Geografia, História, Informática, Educação Física, Biologia, Inglês, totalizando 10 disciplinas, assim como professores fictícios foram criados para a simulação: Abilio, Bianca, Carlos, Denise, Feliciano, Gabriela, Horácio, Indiara, Jandira, Tiago, Paulo, Honória, Sonia, Mário, totalizando 14 profissionais.

Tabela 2 – Configurações da grade para os experimentos realizados

Número de turmas	Horários diários
5	5
10	5
12	5

Tabela 3 – Configurações do SA para os experimentos realizados

Experimento	Temperatura	Número de iterações
1	1000	1000000
2	100e20	10000000
3	100e5	4000000
4	500	2000000

Figura 8 – Grade horária com conflitos em destaque.

Turmas	SEG					TER					QUA					QUI					SEX				
	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
101	83	83	91	94	86	93	85	87	85	96	96	91	92	82	87	89	82	88	90	89	86	92	84	90	84
102	88	87	94	91	86	90	92	86	82	90	87	92	83	93	83	82	96	84	89	85	84	89	96	85	91
103	87	90	84	89	82	91	82	93	90	92	91	83	89	94	87	86	96	85	86	88	92	84	83	96	85
104	82	91	84	96	94	83	93	89	92	89	92	82	86	90	85	84	87	96	86	91	85	88	87	90	83
105	92	82	89	86	96	96	91	90	93	84	90	86	91	83	82	89	84	92	85	94	88	87	83	85	87
106	82	89	91	83	90	87	88	87	85	84	96	92	93	84	90	82	96	83	91	92	86	89	86	94	85
107	83	85	90	93	91	82	86	91	82	96	92	84	83	94	88	92	87	84	96	86	89	85	87	90	89
108	92	91	94	87	83	96	85	88	86	89	93	83	92	87	89	91	85	90	86	84	82	90	84	96	82
109	92	86	91	85	89	86	83	85	90	87	83	82	82	96	94	88	90	84	96	87	93	84	89	91	92
110	82	82	83	83	84	84	85	85	86	86	87	87	88	89	89	90	90	91	91	92	92	93	94	96	96

 **Conflitos**

Quando existe redução no conjunto de dados utilizados para fins de experimento na geração automática da *grade horária* (GH), os dados acima listados são eliminados ordenadamente dos últimos para os primeiros até haver adequação à definição dos parâmetros de grade do experimento. Por exemplo em uma grade que utiliza 5 turmas, excluiu-se através da interface simplificada em um navegador, as turmas de 106 a 112.

Os *slots* definidos com a finalidade de realização dos experimentos simulando uma situação real de elaboração de GH, incluindo os fatores de suas respectivas disciplinas são mostrados na [Tabela 4](#). Podemos observar não haver professor associado a uma disciplina com fator maior que 2, isso se dá, em função da já mencionada formulação no [Item 6](#) do [Capítulo 5](#), por esse motivo existe mais de um professor para a disciplina matemática na mesma grade.

Tabela 4 – Ligações dos professor a suas respectivas disciplina e estas ao fator correspondente. *Slots*

Slot	Professor	Disciplina	Fator
82	Abilio	matemática	2
83	Bianca	portugues	2
84	Carlos	física	2
85	Denise	química	2
86	Edson	biologia	2
87	Feliciano	história	2
88	Gabriela	geografia	1
89	Horácio	inglês	2
90	Indiara	ed.física	2
91	Jandira	informática	2
92	Tiago	matemática	2
93	Paulo	física	1
94	Honória	matemática	1
96	Mario	portugues	2

A [Figura 9](#) mostra a evolução do custo das grades durante o processo de busca para os 10 testes realizados dentro do experimento 1, podemos verificar uma queda rápida do custo seguida por uma tendência a convergência de valores.

O tempo médio de convergência ao valor de custo resultante foi de 3 min e 25 segundos.

Podemos observar pela [Tabela 5](#) que o crescimento do número de variáveis implica em acréscimo nos custos das grades de forma não linear. Demonstrando-se desta forma, o número de turmas como variável de decisão para a busca de soluções (grades horárias) de menor custo, sendo assim quanto mais turmas forem incluídas na grade maior o custo final da mesma.

6.2 Experimento 2

Para o experimento 2 escolheu-se uma parametrização com valores elevados de temperatura e número de iterações, buscando melhorar os resultados obtidos no experimento 1 adotando-se a estratégia contrária ao mesmo.

A tabela [Tabela 6](#) mostra os custos das grades obtidas durante a execução de 10 vezes as configurações do Experimento 2.

Tabela 6 – Resultados para configuração do SA: Temperatura inicial 100e20, total de iterações 10000000, as grade geradas podem ser vistas detalhadamente no Apêndice A.

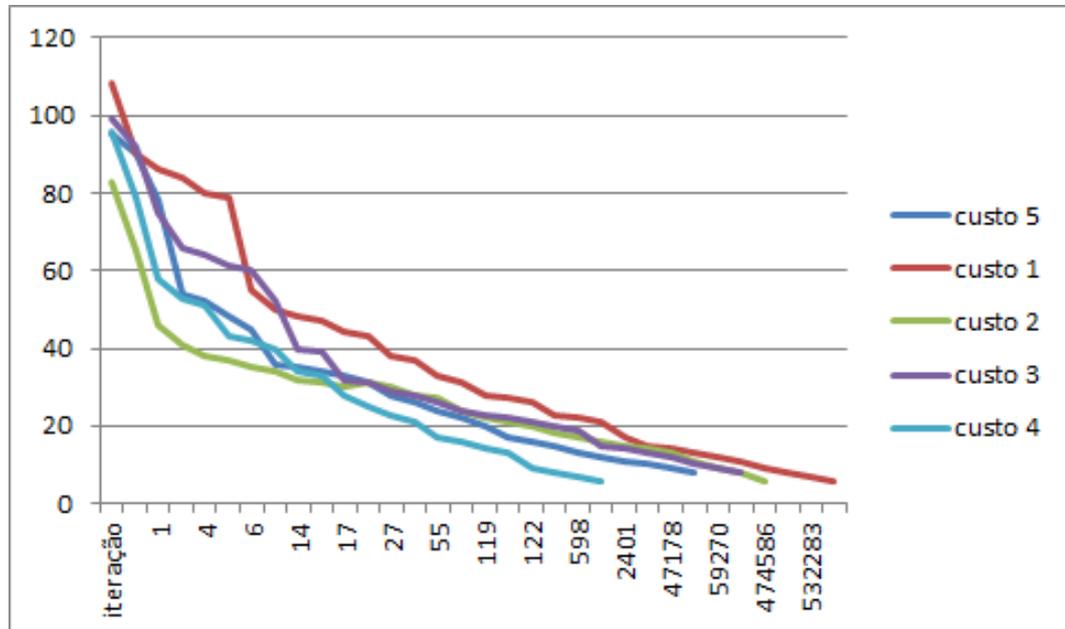
Número de Turmas	1	2	3	4	5	6	7	8	9	10	Média
5	1	0	0	1	2	2	1	1	2	0	1
10	6	6	8	8	6	8	8	6	8	8	7,2
12	51	53	53	54	53	51	52	54	54	52	53,3

A [Figura 10](#) mostra a evolução do custo das grades durante o processo de busca realizado pelo algoritmo SA, podemos verificar uma queda rápida do custo seguida por uma tendência a convergência de valores.

A alteração nos parâmetros do SA demonstraram ser uma alternativa válida para a instância do STP com menor número de variáveis (5 turmas) e demonstraram resultados menos significativos quando o número de variáveis cresce (instâncias com 10 e 12 turmas).

Percebe-se através do [Figura 10](#) que ocorre uma redução rápida do custo das soluções durante as primeiras iterações até chegar em determinados valores onde existe uma convergência a um mínimo local, a melhora do custo da solução passa então a ocorrer mais lentamente.

Figura 10 – Convergência dos custos de 5 grades geradas pela aplicação no experimento 2. Instância de 10 turmas. No eixo x, número ordinal da iteração onde houve atualização do custo da melhor grade até o momento, no eixo y o valor do custo obtido naquela iteração.



A utilização de um número elevado de iterações no experimento 2 não representou um ganho expressivo na redução do custo da solução, considerando que o tempo dispendido para executar tal experimento foi de em média 30 minutos.

6.3 Experimento 3

O principal objetivo para o experimento 3 consiste em obter grades horárias com baixo custo (poucos conflitos), através de uma tentativa empírica de encontrar uma parametrização que obtenha melhores resultados que os experimentos 1 e 2 dentro dos limites dos valores de parâmetros do SA dos mesmos.

A tabela [Tabela 7](#) mostra os custos das grades obtidas durante 10 execuções as configurações do experimento 3.

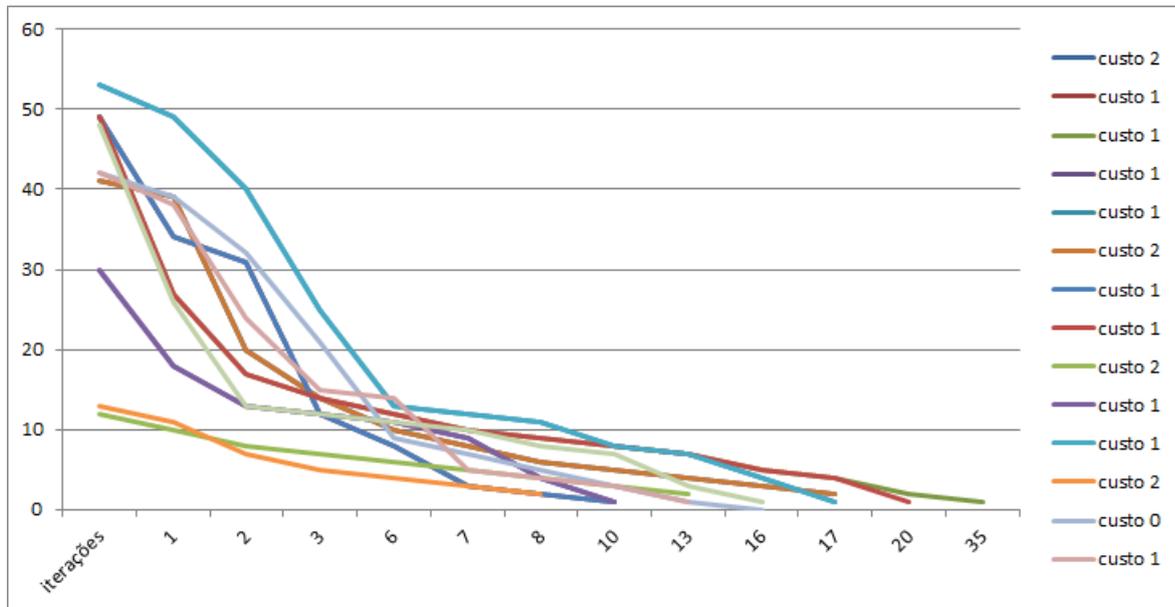
Tabela 7 – Resultados para configuração do SA: Temperatura inicial 100e5, total de iterações 4000000, as grade geradas podem ser vistas detalhadamente no no Apêndice A.

Número de Turmas	1	2	3	4	5	6	7	8	9	10	Média
5	1	1	0	2	1	1	2	1	1	2	1,2
10	8	9	10	11	9	8	10	9	11	10	9,5
12	52	53	55	53	53	51	52	51	54	52	52,6

O [Figura 11](#) mostra a evolução do custo das grades durante o processo de busca

realizado pelo algoritmo SA, podemos verificar uma queda rápida do custo seguida por uma tendência a convergência de valores.

Figura 11 – Convergência dos custos das 10 grades geradas pela aplicação no experimento 3. Instância de 5 turmas. No eixo x, número ordinal da iteração onde houve atualização do custo da melhor grade até o momento, no eixo y o valor do custo obtido naquela iteração.



Os resultados obtidos no experimento 3 demonstraram que a parametrização do SA para o caso médio não representou grande diferença no custo de cada uma das grades geradas.

6.4 Análise dos resultados

Tabela 8 – Resultados de cada instância do STP para todos os experimentos

Número de turmas	exp 1	exp 2	exp 3	Média
5	2,1	1	1,2	1,43
10	9,7	7,2	9,5	8,8
12	53,3	53,3	52,6	53,06

Analisando os resultados obtidos pelos experimentos realizados, inferiu-se as seguintes afirmações relativas ao custo das grades geradas automaticamente: o algoritmo demonstrou-se eficiente para as instâncias do STP com restrições mais folgadas (5 turmas), sendo que soluções ótimas foram obtidas para tais casos, já para instâncias mais restritivas (10 e 12 turmas) não houveram soluções ótimas, atribui-se ao fato de se fazer necessário a utilização de busca local, conforme exposto por (POULSEN, 2012) e (SANTOS

H.G.;OCHI, 2012) uma possibilidade proposta por (BELIGIANIS G.N. ; MOUSCOPOULOS, 2008) seria a hibridização com algoritmos evolutivos .

Não houve praticamente variação no custo das grades geradas para a instância mais restritiva (12 turmas), independente de alteração dos parâmetros do SA.

Como já tem sido exposto nos trabalhos de (POULSEN, 2012), as últimas aulas (últimos períodos da semana) normalmente apresentam os horários mais problemáticos para atribuição sem conflitos devido ao fato de os mesmos contarem com um número menor de possibilidades de atribuição, o autor propôs uma estratégia utilizando estratégias de *backtracking* que soluciona parcialmente tal comportamento. O algoritmo SA apresenta o mesmo comportamento para as últimas aulas da semana.

A convergência de valores em todos os experimentos executados apresentou o mesmo comportamento, rápida redução do custo das soluções nas primeiras iterações seguido na maioria das vezes por um estacionando em algum mínimo local, salvo exceções que representam no melhor caso uma amostra de 20% do total de testes executados para cada experimento.

Outros experimentos foram realizados porém não demonstrarem serem alternativas viáveis para redução no custo da grade gerada, dentre estes podemos destacar: o método conhecido como *reannealing* (usar a saída de uma grade como grade inicial em novo processo de busca), testes com temperaturas realmente baixas na ordem de poucas centenas assim como extremamente altas na ordem de centenas de milhões.

7 Conclusão

Demonstrou-se neste trabalho que mesmo uma versão simplificada do *School Timetabling Problem* (STP), pode ter correspondência com situações vivenciadas por elaboradores de grade horária (GH) em instituições de ensino reais e que uma aplicação web pode ser usada de forma eficiente para automatizar tal processo utilizando o algoritmo *Simulated Annealing* (SA).

Alcançou-se os objetivos propostos relacionados a aplicação web: Inserção de dados eficiente, a interface simplificada serviu bem sua função de simplificar a inserção dos dados dos agentes participantes da grade (turmas, professores, disciplinas...). Tempo de processamento dos dados reduzido, considerando os trabalhos já desenvolvidos na área pode-se classificar de satisfatório o tempo de execução para todos os experimentos realizados. Geração de uma grade consistente com o menor número de conflitos possível como saída, este objetivo foi alcançado conforme demonstrado na [Tabela 8](#) com maior eficiência para as instâncias menos restritivas (5 e 10 turmas), resultados esses semelhantes aos obtidos por trabalhos correlatos como os de ([GANS, 1981](#)) e ([MOURA A.V.; SCARAFICCI, 2004](#)).

A não ser para os casos onde a instância do School Timetabling Problem (STP) apresentava ter restrições folgadas (5 turmas) não foi possível encontrar alguma grade horária com solução ótima, o fato leva-nos a concluir que a melhor maneira de usufruir do algoritmo Simulated Annealing (SA) seja aliando suas características de busca com outra estratégia de busca local, que possibilitem resolver as restrições de atribuição de horário dos últimos períodos da semana, tal abordagem tem sido desenvolvida nos trabalhos de ([BURKE E. K.; PETROVIC, 2002](#)) e ([AVELLA P.; D'AURIA, 2007](#)).

Uma simulação com temperatura alta e um número alto de iterações obtiveram os melhores resultados em termos de custo da grade gerada (número de conflitos). Demonstraram-se portanto serem mais promissoras as simulações que utilizem tal parametrização do SA quando utilizado em uma aplicação web geradora de grade horária automatizada.

O tempo de execução dos experimentos demonstrou ser um dos aspectos a serem melhorados no desenvolvimento de futuras aplicações web geradoras de grades horárias, a média geral para cada experimento em todas as instâncias ficou em torno de 22 minutos.

Como sugestão de trabalhos futuros pode-se mencionar a criação de uma interface amigável (front-end) para a aplicação desenvolvida neste trabalho (back-end). Uma hibridização do SA com uma meta-heurística que não comprometa o desempenho da aplicação web, pode ser uma alternativa a ser explorada. A utilização do XHSTT, quando tal re-

curso esteja amadurecido como estruturador dos dados componentes da grade, também é uma alternativa a ser explorada.

Referências

- ABRAMSON D.; KRISHNAMOORTHY, M. D. H. Simulated annealing cooling schedules for the school timetabling problem. *Asia-Pacific Journal of Operational Research*, 16(1), 1–22., 1999. Citado 2 vezes nas páginas 16 e 23.
- ALVAREZ-VALDEZ R.; CRESPO, E. T. J. Design and implementation of a course system using tabu search. *Journal of operational Research Society*, 137, 512-523., 2002. Citado na página 16.
- ANDREA, S. Survey of automated timetable. *Artificial Intelligence Review*, Kluwer Academic Publisher, 1999. Citado na página 23.
- AVELLA P.; D'AURIA, B. S. S. V. I. A computational study of local search algorithms for italian high school timetabling. *Journal of Heuristics*, 13(6), 543–556., 2007. Citado 2 vezes nas páginas 23 e 45.
- BARDADYM, V. A. Computer-aided school and university timetabling: the new wave. practice and theory of automated timetabling. in: Burke e.; ross, p.(ed.). Lectures Notes in Computer Science, n.1408. Berlin: Springer-Verlag, 1996, p.22-45, 1996. Citado 2 vezes nas páginas 13 e 15.
- BELIGIANIS G.N. ; MOUSCOPOULOS, C. . L. S. A genetic algorithm approach to school timetabling. In: . [S.l.]: In Proceedings of the 7th international conference on the practice and theory of automated timetabling(PATAT2008)., 2008. Citado 3 vezes nas páginas 16, 23 e 43.
- BIRBAS T.; DASKALAKI, S. H. E. School timetabling for quality student and teacher schedules. In: . [S.l.: s.n.], 2008), publisher=Journal of Scheduling, v.12, p.177–197. Nenhuma citação no texto.
- BROOKSHEAR, G. The development of computer science: A sociocultural perspective. University of Joensuu, Department of Computer Science and Statistics, 2003. Citado na página 21.
- BURKE E. K.; PETROVIC, S. Recent research directions in automated timetabling. *European Journal of Operational Research*, v.140, p.266-280,2002., 2002. Citado 2 vezes nas páginas 23 e 45.
- CARRASCO M.P.; PATO, M. V. A comparison of discrete and continuous neural network approaches to solve the class/teacher timetabling problem. In: . [S.l.]: Operational Research, 2004. Nenhuma citação no texto.
- COLORNI A.; DORIGO, M. M. V. *Computation Optimization and Applications*. [S.l.: s.n.], 1998. Citado 2 vezes nas páginas 16 e 23.
- COOPER T.B.; KINGSTON, J. The solution of real instances of the timetabling problem. *The computer journal* 36(7), 645-653., 1993. Citado 2 vezes nas páginas 13 e 23.

- COOPER T.B.; KINGSTON, J. The complexity of timetable construction problems. In Selected papers from the first international conference on the practice and theory of automated timetablingpp. 283-295, 1996. Citado na página 15.
- ELMOHAMED M., C. P. G. A comparison of annealing techniques for academic courses scheduling. In: . [S.l.: s.n.], 1997. Citado na página 16.
- EVEN S.; ITAI, A. S. A. On the complexity of timetable and multicommodity flow problems. In: . [S.l.: s.n.], 1976. Citado na página 16.
- GANS, O. de. A computer timetabling system for secondary schools in the netherlands. In: . [S.l.: s.n.], 1981. Citado 2 vezes nas páginas 23 e 45.
- GOODRICH, M. Algorithms and complexity. In: . [S.l.: s.n.], 2004. Citado na página 21.
- GOTLIEB, C. The construction of class-teacher timetabling. In: . [S.l.: s.n.], 1962. Citado na página 16.
- HANSSON D.H. ; THOMAS, D. Desenvolvimento web Ágil com rails. In: . [S.l.]: The pragmatic programmers. Bookman Companhia Editora, Artmed Companhia Editora. São Paulo, 2007. Citado na página 34.
- JACOBSEN F.; BORTFELDT, A. G. H. Timetabling at german secondary schools: Tabu search versus constraint programming. In: . [S.l.]: Practice and Theory of Automated Timetabling (PATAT 2006), 2007. Nenhuma citação no texto.
- JOHNSON, A. W. J. S. H. S. K. Discrete manufacturing process design optimization using computer simulation and generalizing hill climbing algorithms. *Engeneering Optimization*,31 pag.247-260, 1998. Citado na página 31.
- KIRKPATRICK S. C. D.; GELATT M. P.; VECCHI, S. Optimization by simulated annealing. *Science, New Series*, Vol. 220, No. 4598. (May 13, 1983), pp. 671-680., 1983. Citado na página 29.
- LAWRIE, N. An integer linear programming model of a school timetabling problem. In: . [S.l.]: The computer journal, v.12,n.4, p.307-316, 1969. Nenhuma citação no texto.
- MARTE, M. Towards constraint-based school timetabling. In: . [S.l.]: Annals of Operational Research, v.155, n.1, p.207-225, 2007, 2007. Nenhuma citação no texto.
- METAHEURISTICS 2011. [Http://www.utwente.nl/ctit/hstt/itc2011/welcome/](http://www.utwente.nl/ctit/hstt/itc2011/welcome/). Accessed: 2014-09-30. Citado na página 26.
- MICHALEWICZ Z.;FOGEL, D. *How to solve it: Modern Heuristics, second edition*. 2. ed. Germany: Springer-Verlag Berlin Heidelberg, 2004. Citado na página 31.
- MONTEVECHI J. A. B. ; BATEMAN, R. E. B. R. O. T. J. H. C. R. M. J. R. *Simulação de Sistemas: Aprimorando Processos de Logística, Serviços e Manufatura*. [S.l.]: Campus Elsevier e-book,2002, 2002. Citado na página 22.
- MOURA A.V.; SCARAFICCI, R. A. A strategy for a more constrained school timetabling problem. *International Journal of Operational Research*, 2004. Citado 4 vezes nas páginas 13, 15, 23 e 45.

- OCHI L. S.; SOUZA, M. J. F. S. H. G. An efficient tabu search heuristic for the school timetabling problem. *Lectures Notes in Computer Science*, (vol.3059, pp. 468-481), 2004. Citado 3 vezes nas páginas 16, 23 e 26.
- PILLAY, N. A survey of school timetabling research. Springer Science+Business Media New York 2013 Ann Oper Res DOI 10.1007/s10479-013-1321-8, 2013. Citado 2 vezes nas páginas 21 e 23.
- POST G.;AHMADI, S. S. J. J. C. D. An xml format for benchmarks in high school timetabling. *Annals of Operational Research* 194, 385-397, 2012. Citado na página 26.
- POULSEN, C. J. B. Desenvolvimento de um modelo para o school timetabling problem baseado na meta-heurística simulated annealing. In: . [s.n.], 2012. Disponível em: <<http://www.lume.ufrgs.br/handle/10183/39522-acessadoem01/01/2015>>. Citado 7 vezes nas páginas 7, 15, 16, 25, 31, 42 e 43.
- SANTOS H.G.;OCHI, L. M. N. Strong bounds with cut and column generation for class-teacher timetabling. *Annals of Operational Research*, 194, 399-412, 2012. Citado 4 vezes nas páginas 23, 26, 33 e 43.
- SEVERO L.C.;GIRARDI, A. F. A. M. Simulated annealing to improve analog integrated circuit design: Trade-offs and implementation issues. 2013. Citado na página 29.
- SMITH KATE A.; ABRAMSON, D. D. D. Hopfield neural network for timetabling: formulations methods, and comparative results. In: . [S.l.]: Computer Industrial Engineering, v.44, p.283-305, 2003. Nenhuma citação no texto.
- VALOUXIS CHRISTOS; HOUSOS, E. Constraint programming approach for school timetabling, v.30, p.1555-1572. In: . [S.l.]: Computer Operations Research, 2003. Nenhuma citação no texto.
- WERRA, D. de. Construction of school timetables by flow methods. *Information System and Operational Research*, 9(1), 12-22., 1971. Citado na página 23.
- WRIGHT, M. School timetabling using heuristic search. *Journal of Operational Research Society*, v.47. p.347-357, 1996. Nenhuma citação no texto.
- ZHANG DEFU; LIU, Y. H. R. L. S. Simulated annealing with a new neighborhood structure based algorithm for a high school timetabling problems. In: . [S.l.: s.n.], 2010. Nenhuma citação no texto.

APÊNDICE A – Apêndice A

Tabela 17 – Grade gerada para configuração do experimento 2 - custo da solução: 10

Turma	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5						
101	93	91	88	83	91	96	82	92	94	87	89	82	87	89	86	84	92	90	86	84	84	96	90	85	85	
102	92	86	85	89	83	84	87	94	88	87	90	84	92	96	91	82	89	86	85	91	82	83	90	91	82	96
103	90	82	83	84	88	85	96	84	92	86	96	92	91	94	87	86	86	90	89	82	83	87	89	91	85	85
104	91	89	93	90	84	94	86	86	84	96	82	90	85	82	92	88	87	87	83	85	85	92	83	96	91	85
105	83	87	86	92	82	92	90	89	93	85	86	88	94	87	96	96	83	82	90	91	84	84	85	89	91	85
106	82	96	90	96	89	90	94	85	89	84	84	85	86	83	91	91	93	92	92	87	82	88	87	86	83	83
107	86	85	96	91	87	84	88	83	87	90	92	86	84	91	85	83	89	93	96	90	94	82	82	89	82	82
108	90	83	94	85	86	87	93	87	86	92	85	91	88	92	84	82	91	84	89	96	96	89	82	83	84	84
109	85	88	82	89	96	83	92	90	96	91	83	87	89	86	86	87	82	91	85	93	90	94	84	84	92	92
110	96	92	89	87	90	89	84	82	91	83	88	83	96	93	84	94	85	90	84	84	92	85	91	86	87	87
conflitos	1	0	0	1	0	1	0	0	0	1	0	0	0	0	0	0	0	1	0	0	1	0	0	2	2	2

Tabela 18 – Grade gerada para configuração do experimento 2 - custo da solução: 9

Turma	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5					
101	87	83	96	90	86	87	85	91	96	93	90	82	86	92	85	89	82	88	94	89	84	83	84	91	92
102	82	96	87	83	82	92	86	87	90	88	96	91	83	84	90	89	86	85	93	94	91	92	89	84	85
103	84	87	92	85	89	91	83	82	82	86	91	94	85	88	84	90	92	93	89	96	90	86	87	96	83
104	92	89	88	94	91	96	84	96	85	85	89	84	92	93	86	83	87	90	82	82	86	91	90	83	87
105	96	91	85	89	87	83	94	92	84	88	85	89	96	86	91	87	90	86	83	84	93	90	92	82	82
106	89	82	96	82	90	84	90	93	87	91	86	96	88	83	87	92	85	91	84	85	92	94	83	89	86
107	90	84	82	92	84	85	82	90	94	89	93	86	91	91	88	85	96	92	86	83	87	83	96	87	89
108	86	94	90	86	92	90	89	91	83	83	82	87	84	85	93	84	91	87	88	92	89	96	85	82	96
109	85	85	91	87	96	94	92	89	91	92	88	83	87	90	82	86	89	82	96	93	83	84	86	90	84
110	93	90	86	96	83	92	91	83	89	94	84	92	82	87	89	88	84	96	91	90	82	87	85	85	86
conflitos	0	0	1	0	0	1	0	1	0	1	0	0	0	0	0	1	0	0	0	0	0	1	1	1	1

Tabela 19 – Grade gerada para configuração do experimento 2 - custo da solução: 9

Turma	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5					
101	87	83	96	90	86	87	85	91	96	93	90	82	86	92	85	89	82	88	94	89	84	83	84	91	92
102	82	96	87	83	82	92	86	87	90	88	96	91	83	84	90	89	86	85	93	94	91	92	89	84	85
103	84	87	92	85	89	91	83	82	82	86	91	94	85	88	84	90	92	93	89	96	90	86	87	96	83
104	92	89	88	94	91	96	84	96	85	85	89	84	92	93	86	83	87	90	82	82	86	91	90	83	87
105	96	91	85	89	87	83	94	92	84	88	85	89	96	86	91	87	90	86	83	84	93	90	92	82	82
106	89	82	96	82	90	84	90	93	87	91	86	96	88	83	87	92	85	91	84	85	92	94	83	89	86
107	90	84	82	92	84	85	82	90	94	89	93	86	91	91	88	85	96	92	86	83	87	83	96	87	89
108	86	94	90	86	92	90	89	91	83	83	82	87	84	85	93	84	91	87	88	92	89	96	85	82	96
109	85	85	91	87	96	94	92	89	91	92	88	83	87	90	82	86	89	82	96	93	83	84	86	90	84
110	93	90	86	96	83	92	91	83	89	94	84	92	82	87	89	88	84	96	91	90	82	87	85	85	86
conflitos	0	0	1	0	0	1	0	1	0	1	0	0	0	0	0	1	0	0	0	0	0	1	1	1	1

Tabela 23 – Grade gerada para configuração do experimento 3

Turma	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5						
101	87	83	92	92	99	98	82	89	91	87	91	83	86	85	90	89	85	84	93	99	90	88	84	82	86	
102	89	84	83	86	82	92	90	87	89	91	99	82	99	83	98	90	84	85	87	88	92	86	85	93	85	
103	90	99	89	82	86	85	86	85	99	98	92	91	84	87	87	83	88	93	92	90	82	83	84	91	89	
104	98	91	85	84	92	83	84	82	87	85	90	89	88	89	91	93	86	99	90	92	99	87	83	86	82	
105	92	85	88	99	87	86	83	98	90	83	93	92	85	91	84	84	89	86	82	82	91	90	99	87	89	
106	91	82	99	83	98	84	93	91	82	86	85	90	87	90	89	99	92	87	85	86	88	92	89	83	84	
107	99	92	84	87	89	91	91	90	86	82	83	87	93	82	92	85	99	89	84	83	98	88	86	85	90	
108	93	89	86	98	83	99	85	84	92	90	84	88	89	86	99	82	90	83	91	91	87	82	87	92	85	
109	83	87	93	85	84	90	92	83	84	92	86	99	82	88	85	86	82	90	89	87	89	91	98	91	99	
110	84	90	91	93	90	87	88	92	98	99	91	84	92	99	83	87	83	82	86	89	85	89	82	86	85	
conflitos	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	1	1	2	3