# Heurísticas eficientes para o problema de geração de grade escolar automatizada

**Eraldo Paravidino Neto**
eraldoneto@gmail.com

**Dalessandro Soares Vianna**
dalessandrosoares@yahoo.com.br

**RESUMO**

*As instituições de ensino têm grande dificuldade no momento da geração da grade escolar. A programação de horários torna-se tarefa difícil devido ao grande número de possibilidades e à necessidade de se respeitar uma série de restrições, muitas vezes conflitantes entre si. Podem-se levar semanas ou, até mesmo, meses para se conseguir uma solução satisfatória. Além disso, os resultados normalmente causam insatisfação tanto dos professores quanto dos alunos. Uma solução que seja capaz de propor uma grade com qualidade e que não viole as restrições é muito necessária, uma vez que é crescente o número de novos estabelecimentos de ensino e o problema estar presente em todos eles. O uso de heurísticas para geração automática da grade visa facilitar o trabalho e, ao mesmo tempo, buscar uma solução que melhor atenda às expectativas da instituição. Foi feito um comparativo entre a qualidade das soluções encontradas utilizando um algoritmo GRASP e um ILS para determinar o melhor algoritmo a ser usado na busca de solução. Os resultados indicaram que, dependendo do conjunto de disponibilidades dos professores a ser utilizado, um algoritmo pode convergir para melhores soluções mais rápidas que o outro algoritmo.*

***Palavras-chave:*** *Grade escolar; Heurística; GRASP; ILS.*

## Efficient heuristics for the problem of generating automated school timetables

**ABSTRACT**

*Teaching institutions have great difficulty, at the moment, of generating school timetable. Planning the timetable becomes a hard task due to the great number of possibilities and to the need to respect a series of restrictions, which are, many times, conflicting with each other. It can take weeks or even months to achieve a satisfactory solution. Besides, the results generally cause dissatisfaction both to teachers and students. A solution which is able to propose a quality curricular matrix and which does not violate the restrictions is very necessary, since there is an increasing number of new teaching institutions and the problem belongs to all of them. The use of heuristics to generate an automated school timetable has the objective to facilitate the work and, at the same time, look for a solution which best assists the institution's expectations. A comparison between the quality of the solutions found was done, using a GRASP and an ILS algorithm, to determine the best algorithm to be used in search for a solution. The results indicate that, depending on the number of teachers' to be used, an algorithm can produce better and quicker solutions than the other algorithm.*

***Key words:*** *School timetable; Heuristics; GRASP; ILS.*

## 1. Introdução

A geração da grade escolar sempre é uma grande preocupação para as instituições de ensino. O processo é muito lento e difícil, podendo demorar semanas ou até meses para se encontrar uma solução, ainda que gerada por mais de um funcionário. Além disso, essa solução, muitas vezes, não atende às expectativas das partes envolvidas nesse processo. Podem-se destacar como partes envolvidas o professor com suas disponibilidades de horários e as turmas que necessitam de determinado conjunto de disciplinas, sendo para cada disciplina necessária uma quantidade de aulas a ser dada.

Além das partes envolvidas, é necessário que se leve em consideração que cada instituição possui um conjunto de restrições que devem ser consideradas no momento da geração da solução. Esse conjunto de restrições muda de acordo com cada instituição, fazendo com isso que a criação de um algoritmo adequado para a geração de uma grade escolar que atenda aos desejos da instituição seja mais difícil de ser desenvolvido.

Visto isso, um algoritmo que gere uma solução que atenda a todos os requisitos ou, pelo menos, a maior parte deles é muito necessária para qualquer instituição. Este artigo propõe dois algoritmos para resolução do problema: GRASP e ILS. Foi feito um comparativo entre as soluções encontradas de acordo com o conjunto de dados iniciais disponibilizados. A partir desse comparativo, fez-se uma avaliação para determinar quando é aconselhável usar o GRASP e quando usar o ILS, tendo como base a curva de melhoria da qualidade da solução gerada por cada um deles.

Este artigo está organizado da seguinte forma: Na seção 2, descreve-se o objetivo do projeto; na seção 3, o problema que motivou a criação do projeto; na seção 4, define-se o algoritmo GRASP; na seção 5, o algoritmo ILS; na seção 6, são mostrados os resultados; e, por fim, na seção 7, faz-se a conclusão do projeto.

## 2. Objetivo

O objetivo deste artigo foi propor dois algoritmos que possibilitem a instituição gerar automaticamente a grade escolar. A partir das soluções encontradas pelos algoritmos em tempo pré-fixado, é feito um comparativo da qualidade da solução encontrada por cada um deles, pelo tempo que foi preciso para encontrá-la. De posse desse comparativo, faz-se uma análise de quando é aconselhável usar um ou outro algoritmo.

A grande motivação para a criação de um gerador automático é devida ao fato de a geração de uma grade escolar de forma manual demandar muito esforço e tempo para se chegar a uma solução que tente atender, da melhor forma possível, às necessidades da instituição. O problema é que na maioria das vezes as soluções encontradas por melhor que sejam feitam ou, por maior esforço que se tenha gasto, não satisfaz as necessidades de todas as pessoas envolvidas.

O tempo que seria gasto de um ou mais funcionários para o desenvolvimento da geração manual pode ser empregado para atender a outras necessidades da instituição. Com isso, libera-se o funcionário de uma atividade que se torna cansativa e complexa demais, pois envolve um série de requisitos que devem ser atendidos, o que para um ser humano demanda muito esforço, posto ser necessário considerar todos os requisitos pretendidos pela instituição ao mesmo tempo no momento da criação da grade.

Há grande variedade de abordagens desenvolvidas por pesquisadores para resolver as diferentes instâncias dos problemas de grade escolar, o qual é do tipo NP-difícil (COOPER; KINGSTON, 1996), devido à gama de particularidades que podem existir em cada instituição. Como exemplo de particularidades que podem ocorrer, pode-se citar uma instituição onde os estudantes podem escolher suas matérias ou existem aulas com durações diferentes, entre tantos outros exemplos. Isso faz que a definição de um algoritmo que possa atender a essas particularidades seja complexa de ser desenvolvida.

Visando minimizar esse esforço e buscar soluções melhores que satisfaçam uma gama maior de pessoas envolvidas, são usados os algoritmos GRASP (*Greedy Randomized Adaptive Search Procedure*) e ILS (*Iterated Local Search*), para gerar soluções de melhor qualidade para a instituição. A partir das soluções encontradas, é feita uma comparação para determinar qual traria a melhor solução em um tempo pré-fixado, de acordo com os dados iniciais disponibilizados.

## 3. Descrição do Problema

O problema da geração de grade escolar consiste na relação entre os professores que

lecionarão as aulas e as turmas para as quais serão ministradas. Cada turma está associada a um período (1º, 2º ou 3º) da instituição e a um turno (manhã, tarde ou noite). Nenhuma turma estará associada a mais de um período ou a mais de um turno. Cada professor possui um conjunto de disciplinas que está habilitado a lecionar, além de um conjunto de horas disponíveis para tal. O conjunto de horas é previamente informado ao professor para ser considerado durante a geração da solução. Para a instituição à qual está sendo feita a geração da grade, os professores informam 40 h antes que estarão disponíveis durante a semana, distribuídas entre os turnos (manhã, tarde e noite), pois a instituição em questão possuía turmas em todos esses turnos. Com relação às turmas, cada uma delas tinha um conjunto de disciplinas necessárias, de acordo com o período e turno delas. Somado a isso, cada disciplina necessária para a turma possuía uma quantidade de aulas que precisavam ser dadas. Para a geração da grade escolar automatizada, além da relação de professores, turmas e suas particularidades, é considerado um período de tempo previamente fixado, normalmente de uma semana, além de também satisfazer um conjunto de outras restrições que sãos especificadas pela instituição para que a solução encontrada possa atender a todos os requisitos do problema.

A criação de um gerador automatizado de grade escolar é necessária, pois, segundo Souza (2000), a criação de uma solução manual para esse problema, além de ser tarefa extremamente complexa, pode requerer várias semanas de trabalho. Além disso, a solução obtida pode não satisfazer o problema em diversas restrições necessárias na instituição.

Alguns autores, contudo, entre eles Schaerf (1999) e Werra (1985), acreditam que os problemas de geração de horários não podem ser completamente automatizados. A crença deles se baseia em duas justificativas. A primeira é que há razões que não podem ser facilmente expressas em um sistema automizado, ou seja, existem certos detalhes que devem ser considerados no momento da avaliação da solução, que dificilmente se consegue expressar via código. A outra justificativa é de que, por conta do espaço de soluções que é vasto, a intervenção humana pode conduzir à busca em direção a regiões promissoras, às quais o sistema, por si só, dificilmente teria condições de chegar. Assim, Schaerf (1999) e Werra (1985) acreditam que, para que seja encontrada uma solução que mais se aproxime das expectativas do usuário, é necessário que em certos momentos da criação da solução haja intervenção humana para guiar essa busca.

Normalmente, as escolas atendem a determinado número de turmas que são conjuntos de estudantes que estão no mesmo período escolar. As turmas são distribuídas pelos turnos do dia (manhã, tarde e noite). Não é obrigatório que a quantidade de turmas em cada período seja igual, nem que tenha turmas em todos os períodos. Há variedade nessa distribuição de turmas de escola para escola.

Com relação às turmas, há um conjunto específico de disciplinas que são necessárias que sejam dadas. Para cada disciplina há determinado número de aulas. As disciplinas e respectivas quantidades de aulas variam de acordo com o currículo do curso e o período a que a turma pertence. O caso mais comum é a quantidade de aulas de cada turma preencher completamente a semana, isto é, as turmas têm aulas em todos os horários de seu período, todos os dias da semana. Há casos, contudo, em que podem ocorrer janelas também no horário da turma (CISCON et al., 2005).

As aulas são ministradas por um conjunto de professores. Cada professor tem seu conjunto de disciplinas que ele pode dar aula de acordo com o período das matérias, além de cada professor definir quais horários que o docente esteja disponível para lecionar, pois, muitas vezes, os professores trabalham em mais de uma escola e em cada uma delas ministram diferente número de aulas (CISCON et al., 2005).

Com isso, o objetivo deste trabalho foi criar uma grade escolar que atendesse às seguintes premissas:

- Que a carga horária de todas as matérias seja atendida.
- Que cada turma não tenha mais de uma aula ao mesmo tempo.
- Uque um professor não dê aula para mais de uma turma no mesmo horário.
- Seja respeitada, o máximo possível, a lista de horários disponibilizados pelo professor no momento da geração da grade escolar.

Além de atender a essas premissas, este trabalho buscou atender às observações feitas por vários pesquisadores, citando que muitas vezes um algoritmo criado para gerar a grade escolar não consegue chegar a uma solução realmente satisfatória. Isso porque, como visam sempre buscar a melhor solução, se chega a determinado ponto e não se

consegue mais melhorar a solução encontrada.

Para isso foram desenvolvidos dois algoritmos, em que um deles sempre buscou melhorar a solução e o outro tendeu a piorar um pouco a solução encontrada para depois melhorá-la, procurando, assim, caminhos alternativos para se encontrar uma solução que melhor atenda ao problema.

## 4. GRASP

O algoritmo GRASP (*Greedy Randomized Adaptive Search Procedure*), que em português significa "Procedimento de Busca Gulosa Adaptativa Aleatória", é um algoritmo comumente aplicado a problemas de otimização combinatória.

Ele é um processo interativo, em que cada iteração consiste em duas fases:

- Fase de construção.
- Fase de busca local para melhorar a qualidade da solução.

As iterações GRASP são independentes, ou seja, na iteração atual não se leva em conta nenhuma informação das iterações anteriores.

O critério de parada normalmente usado é um número máximo de iterações. A melhor solução obtida no final da execução do GRASP é a solução final.Na Figura 1 é apresentado o pseudocódigo do algoritmo.

**Enquanto** (condição de parada não for satisfeita) **faça**;
    Construa uma solução *s*;
    Refine s utilzando uma busca local. Seja *s*' a solução refinada;
    **Se** *s*' é a melhor solução até então conhecida **então**;
        Armazene *s*';
    **fim-se**;
**fim-enquantob**.

Figura 1 - Algoritmo GRASP.

A criação da solução é um diferencial para os outros tipos de algoritmos. Ela se baseia nas três primeiras iniciais da sua sigla em inglês: gulosa (*Greedy*), aleatória (*Randomized*) e adaptativa (*Adaptive*).

Nos outros algoritmos, a ênfase é dada na busca local, enquanto o GRASP foca, principalmente, na construção de uma solução de qualidade, utilizando a busca local apenas para pequenas melhorias.

Na fase de construção, a solução é construída iterativamente elemento por elemento, até que a solução esteja completa. Os elementos candidatos que compõem a solução são ordenados em uma lista, chamada de lista de candidatos, a qual contém todos os candidatos. Essa lista é ordenada por uma função gulosa, que mede o benefício que o elemento escolhido mais recentemente concede à parte da solução já construída. Um subconjunto denominado lista restrita de candidatos (LRC) é formado pelos melhores elementos que compõem a lista de candidatos.

A heurística é adaptativa porque os benefícios associados com cada elemento são atualizados a cada iteração da fase de construção para refletir as mudanças ocorridas pela seleção de elementos anteriores. A parte aleatória corresponde à forma de escolha dos melhores candidatos da lista. Cada iteração é composta por três etapas:

- Construção da Lista Restrita de Candidatos (LRC), a qual contém um conjunto reduzido de elementos candidatos a pertencer à solução.
- Escolha aleatória do elemento na LRC e inclusão de elemento na solução.
- Adaptação ou recálculo da função gulosa para os elementos ainda não pertencentes à solução.

Os métodos de busca local em problemas de otimização constituem uma família de técnicas baseadas na noção de vizinhança, ou seja, são métodos que percorrem o espaço de pesquisa passando, iterativamente, de uma solução para outra que seja sua vizinha.

A fase de busca local de GRASP aproveita, assim, a solução inicial da fase de construção e explora a vizinhança ao redor dessa solução. Se um melhoramento é encontrado, a solução corrente é atualizada e novamente a vizinhança ao redor da nova solução é pesquisada. O processo se repete até nenhum melhoramento ser encontrado.

De acordo com isso, é preciso ter cuidado ao:

- Escolher uma vizinhança apropriada.
- Usar estrutura de dados eficientes para acelerar a busca local.
- Ter uma boa solução inicial (perto de uma ótima local), para conduzir a uma busca local eficiente.

Depois da busca local, é verificado se a solução encontrada é melhor que a solução encontrada anteriormente, caso seja a nova solução definida como a solução ideal; caso não seja, ela é descartada.

## 5. ILS

O algoritmo ILS (*Iterated Local Search*),

que significa em português "Busca Local Iterativa", é uma meta-heurística que busca encontrar um bom resultado através de um caminho pelos ótimos locais. Baseia-se na ideia de que se pode melhorar um procedimento de busca local com a geração de novas soluções obtidas com a aplicação de perturbações sobre a melhor solução encontrada até o momento. É um método que não explora o espaço completo das soluções, mas um pequeno subespaço composto por soluções que são ótimos locais ou, ainda, soluções que sofreram uma busca local, mas sem a garantia de otimalidade de determinados procedimentos de otimização (ANDRADE et al., 2008).

Basicamente, a partir da solução corrente, *s*, que sofreu refinamento usando busca local, o que gerou uma nova solução *s'*, ele faz mutações, ou perturbações, gerando uma solução *s''*. Após a geração dessa nova solução é executada uma busca local, gerando-se uma solução *s'''*. Caso essa solução seja melhor, de acordo com uma função de aceitação, que a solução *s''*, ela passa a ser a corrente e, caso ela seja a melhor solução até o momento, é armazenada, e o processo é novamente executado até que uma condição de término da perturbação seja alcançada. Ao fim da condição de término da perturbação, o processo é novamente executado até atingir a condição de parada de execução do algoritmo. Assim, as principais características do ILS são: a perturbação usada, a busca local e o critério de aceitação definido. Na Figura 2 é mostrado o pseudocódigo do algoritmo.

Construa uma solução s;
Refine s utilizando uma busca local. Seja s' a solução refinada;
**Enquanto** (condição de parada não for satisfeita) **faça**;
  Seja s'' a solução refinada s';
  **Enquanto** (condição de parada da perturbação não for satisfeita) **faça**;
    Perturbe a solução s''. Seja s''' a solução perturbada;
    Refine a solução s'''.
    **Se** s''' é uma melhor solução que s'' **então**;
      Armazene s''' em s'';
      **Se** s''' é a melhor solução até então conhecida **então**;
        Armazene s''';
      **fim-se**;
    **fim-se**;
  **fim-enquanto**;
f**im-enquanto**.

Figura 2 - Algoritmo ILS

O potencial da busca local iterativa se encontra em sua amostragem de locais otimizados. Essa amostragem depende dos tipos de perturbações e dos critérios da aceitação. Naturalmente, as perturbações têm efeito na velocidade da busca local: os valores das perturbações fracas conduzem, geralmente, a uma execução mais rápida da busca local. Outros resultados podem ser obtidos também a partir da modularização de componentes no algoritmo, adaptando-se as perturbações, trabalhando com vizinhos mais próximos, esquemas complexos de perturbações, características específicas dos problemas e critérios de aceitação.

O critério da perturbação quando combinado com a busca local ajuda o algoritmo a escapar de mínimos locais e explorar regiões que contenham melhores soluções. A perturbação forte quando aplicada resulta em soluções melhores com probabilidades mais baixas. Quando a perturbação é baixa, o algoritmo frequentemente cai em mínimos locais. Como se pode perceber, a perturbação deve ser forte o suficiente para garantir a exploração de diferentes soluções e fraca o bastante para que não ocorra reinício aleatório. A eficiência do ILS é baseada no conjunto de amostragem de ótimos locais e na escolha do método de busca local, das perturbações do critério de aceitação. Os seguintes procedimentos são utilizados neste trabalho nas etapas que compõem o ILS:

- Geração da solução inicial.
- Busca local.
- Perturbação.
- Critério de aceitação.

Como no algoritmo GRASP, a geração da solução inicial é feita de forma iterativa, elemento por elemento, até que a solução esteja completa. É gerada uma lista de elementos canditados, ordenados através de uma função gulosa que mede o benefício que o elemento escolhido mais recentemente concede à parte da solução já construída. A partir dessa lista é feito um subconjunto, denominado lista restrita de candidatos (LRC), que é formado pelos melhores elementos que compõem essa lista.

O método de busca local também é feito da mesma forma que o algoritmo GRASP, ou seja, ele percorre o espaço de pesquisa passando, iterativamente, de uma solução para outra que seja sua vizinha. Assim, a fase de busca local aproveita a solução inicial e explora a vizinhança ao redor dessa solução. Se um melhoramento é encontrado, a solução corrente é atualizada, e novamente a

vizinhança ao redor da nova solução é pesquisada. O processo se repete até nenhum melhoramento ser encontrado.

Com relação à etapa de perturbação, são efetuadas duas trocas de horários de aulas aleatoriamente. Para cada troca é escolhida uma turma aleatória, e a partir dela escolhem-se duas aulas e troca-se o horário de uma aula com a outra. Após essa perturbação, é efetuada uma busca local nessa nova solução e, caso a solução após a busca local seja a melhor encontrada, ela é definida como solução ideal.

Por fim, o critério de aceitação da solução é baseado na quantidade de infrações cometidas por ela. Quanto menor o número de infrações, melhor a solução encontrada.

## 6. Resultados encontrados

Definidos os algoritmos que são utilizados e implementados, é necessário verificar a qualidade dos algoritmos construídos e validar a confiabilidade das soluções encontradas. Para isso são utilizados dois conjuntos de dados distintos:

- Disponibilidade aleatória dos professores.
- Disponibilidade real dos professores.

Com relação à disponibilidade aleatória, é importante observar que, no momento da geração dos dados, não foi considerada a quantidade de aulas que o professor precisa lecionar por turno, podendo, então, ter sido definidos menos horários disponíveis por turno do que o professor realmente necessitava. Um exemplo da situação que pode ocorrer será o professor precisar lecionar 15 aulas no turno da manhã, porém podem ter sido definidos somente 12 horários disponíveis do professor para esse turno.

Para o conjunto de dados referentes à disponibilidade real dos professores, foi avaliada a quantidade de aulas a serem lecionadas por turno para cada professor, bem como disponibilizados horários de acordo com essa avaliação, ou seja, se o professor for lecionar mais aulas durante o turno da tarde, foram disponibilizados mais horários para esse turno.

Para avaliar a qualidade das soluções, foram somadas todas as quantidades de vezes que a solução infrigiu cada uma das restrições definidas pela instituição e, quanto menor o valor da soma, melhor a qualidade da solução.

Visando ter uma base mais confiável de resultados para os nossos testes, foram efetuadas 10 rodadas destes para cada algoritmo. Cada rodada dessas teve duração de 5 h para não favorecer nenhum dos algoritmos. Além disso, foram feitas as mesmas rodadas de testes para cada conjunto de dados definidos anteriormente.

Na Figura 3, mostra-se o comparativo da solução gerada pelos dois algoritmos utilizando os dados aleatórios de disponibilidade dos professores.

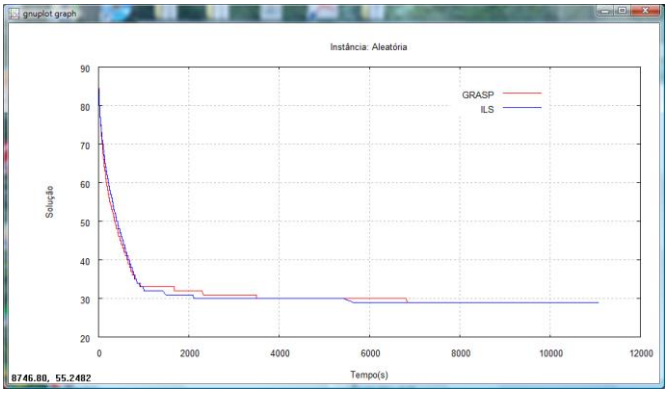


Figura 3 - Quadro comparativo das soluções encontradas pelo algoritmo GRASP e ILS com dados aleatórios

De acordo com o gráfico da Figura 3, pode-se perceber que o algoritmo GRASP teve leve tendência a encontrar uma solução de maior qualidade mais rapidamente que o algoritmo ILS.

Na Figura 4 é mostrado o comparativo da solução gerada pelos dois algoritmos utilizando os dados reais fornecidos pela instituição.

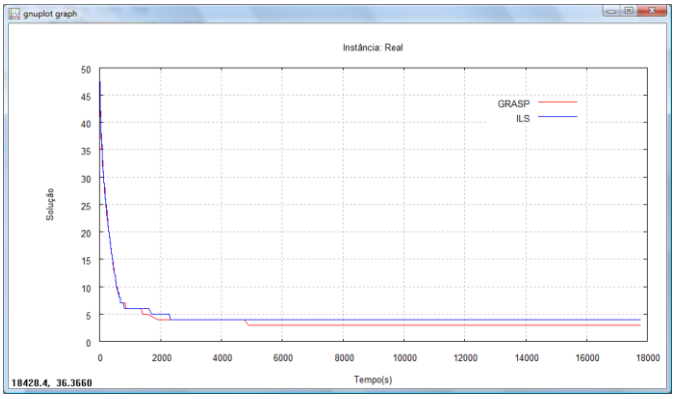


Figura 4 - Quadro comparativo das soluções encontradas pelo algoritmo GRASP e ILS com dados reais

Como é possível observar, o gráfico da Figura 4 apresenta semelhança para o outro, pois nesse o algoritmo GRASP também apresenta leve tendência a achar uma solução de maior qualidade mais rapidamento que o algoritmo ILS.

Apesar de nos dois comparativos o algoritmo GRASP apresentar melhores resultados mais rapidamente que o algoritmo ILS, não é possível afirmar que ele é o melhor algoritmo, pois os resultados foram bem próximos. Além disso,

alguns resultados encontrados pelo algoritmo ILS não foram alcançados em nenhuma das rodadas efetuadas pelo algoritmo GRASP.

Assim, os resultados indicaram que não é possível determinar qual algoritmo é melhor para o problema de geração de grade escolar. Porém, é possível verificar que, quando se utilizam dados mais próximos da realidade da instituição e, ao mesmo tempo, esses dados visam atender a relação entre a quantidade de aulas por turno que o professor lecionará e a quantidade de horários disponibilizados do professor por turno, o algoritmo ILS apresenta leve vantagem. Isso porque o seu algoritmo tende a exibir leves melhorias na qualidade da solução, pois efetua perturbações na solução ótima encontrada, podendo ir para caminhos que facilitem a obtenção de melhores soluções que seriam mais difíceis de serem encontradas com o algoritmo GRASP.

## 7. Conclusões

A utilização de algoritmos para geração automática da grade escolar visa facilitar a instituição de ensino, fazendo que uma solução de qualidade seja encontrada de forma eficiente. Com isso, evita-se que sejam alocados funcionários para um trabalho cansativo e complexo, desnecessariamente. Além disso, os funcionários demorariam muito tempo para encontrar uma boa solução, e na maioria das vezes a solução encontrada não atenderia às expectativas da instituição.

A grande preocupação que se deve ter na criação dos algoritmos é com relação às restrições da instituição. É extremamente importante que no momento da criação da solução sejam consideradas todas as restrições e buscado atendê-las da melhor maneira possível, dentro das possibilidades que o conjunto de dados da instituição oferece. Um exemplo de restrição que pode não ser atendida por causa do conjunto de dados da instituição seria um professor que necessite lecionar mais aulas em determinado turno do que os horários que ele definiu como disponíveis. Essa preocupação com as restrições é essencial para a busca de uma solução de qualidade.

Há vários algoritmos que visam solucionar o problema da geração de grade escolar. Cada algoritmo implementado busca atender à instituição para o qual ele foi criado e, na maioria das vezes, não é possível usar o mesmo algoritmo para outra instituição, devido ao fato de as restrições impostas pela instituição estarem intrínsecas dentro do algoritmo.

Neste artigo foram propostos dois diferentes algoritmos que podem ser adequados para solucionar o problema de geração de grade escolar de outras instituições. A partir da criação desses algoritmos foi feito um comparativo da qualidade da solução encontrada pelo tempo que foi levado na obtenção da solução para determinar qual algoritmo mais se encaixaria na resolução do problema.

De acordo com os resultados, percebe-se que pode ser usado qualquer um dos dois algoritmos, pois eles retornaram soluções de ótima qualidade, porém o algoritmo ILS apresentou leve tendência a encontrar soluções melhores que o algoritmo GRASP. Isso se deve ao fato de que o ILS procura em regiões a que o GRASP pode não chegar, pois ele sempre busca melhorar a solução encontrada, enquanto o ILS realiza perturbações na solução encontrada para realizar outra busca local e, com isso, é possível obter soluções que o GRASP não consegue alcançar.

## 8. Referências

ALVARENGA, F. V.; ROCHA, M. L. Uma metaheurística GRASP para o problema da árvore geradora de custo mínimo com grupamentos utilizando grafos fuzzy. **Jornada da Ciência da Computação**, v. 5, Lavrars, mar. 2006.

ANDRADE, M. S. F.; SOUZA, S. R.; TEMPONI, E. C. C.; SOUZA, M. J. F. Um algoritmo evolutivo híbrido aplicado à solução do problema de corte bidimensional guilhotinado. In: SIMPÓSIO BRASILEIRO DE PESQUISA OPERACIONAL, 40., 2008, João Pessoa. **Anais...** João Pessoa, 2008.

ARAÚJO, T. M. U. de; CABRAL, L. dos A. F.; NASCIMENTO, R. Q. do. Hibridizando a metaheurística c-grasp com o método de busca por padrões adaptativos para resolução de problemas de otimização global contínua. In: SIMPÓSIO DE ENGENHARIA DE PRODUÇÃO, 15., 2008, Bauru, SP. **Anais...** Bauru, SP, 2008.

BIRBAS, T.; DASKALAKI, S.; HOUSOS, E. D de. School timetabling for quality student and teacher schedules. **Journal of Scheduling**, v. 12, Issue 2, Aapr. 2009.

BURKE, E. K.; NEWALL, J. P.; WEARE, R. F. A simple heuristically guided search for the timetable problem. In: THE INTERNATIONAL ICSC SYMPOSIUM ON ENGINEERING OF INTELLIGENT SYSTEMS, 1998, Laguna. **Proceedings...** Laguna: Univ. of La Laguna, 1998.

CISCON, L. A. et al. O problema de geração de horários: um foco na eliminação de janelas e aulas isoladas. In: SIMPÓSIO BRASILEIRO DE PESQUISA OPERACIONAL, 38., 2005, Gramado. **Anais...** Gramado, RS, 2005.

COOPER, T. B.; KINGSTON, J. H. The Complexity of Timetable Construction Problems. In: BURKE, E. K.; ROSS, P. (Ed.). **Practice and Theory of Automated Timetabling, Lecture Notes in Computer Science, Springer-Verlag**, Berlin, v. 1153, p. 283-295, 1996.

SCHAERF, A. A survey of automated timetabling. **Artificial Intelligence Review**, v. 13, p. 87-127, 1999.

SOUZA, M. J. F. **Programação de horários em escolas**: uma aproximação por metaheurísticas. 2000. Tese (Doutorado em Engenharia de Sistemas e Computação) – Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2000.

WERRA, D de. An introduction to timetabling**. European Journal of Operational Research Society**, v. 19, p. 151-162, 1985.

Artigo selecionado entre os 10 melhores do VII Encontro Mineiro de Engenharia de Produção - EMEPRO 2011.