



<a href="#">Novidades</a>	<a href="#">Docente</a>	<a href="#">Locais e horários</a>	<a href="#">Objetivos</a>	<a href="#">Programa da disciplina</a>	<a href="#">Referências bibliográficas</a>	<a href="#">Material didático</a>	<a href="#">Avaliação</a>	<a href="#">Listas de exercícios</a>	<a href="#">Datas importantes</a>
---------------------------	-------------------------	-----------------------------------	---------------------------	--	--	-----------------------------------	---------------------------	--------------------------------------	-----------------------------------

● **Novidades:** [Consulte esta seção frequentemente.](#)

1. Primeira aula em 02/09/2014.

● **Docente:**

[Cid Carvalho de Souza](#)

Sala: 8 (Prédio IC1)

Email para contato: [cid@ic.unicamp.br](mailto:cid@ic.unicamp.br)

**Nota:** nenhum email enviado de endereço de fora do IC será respondido.

**Monitor:** Lucas Oliveira ( [oliveiralukas@yahoo.com.br](mailto:oliveiralukas@yahoo.com.br) , PED)

● **Dias e locais das aulas e do atendimento:**

- **aulas:** as aulas serão nas terças das 14:00 às 16:00 (sala CB06) e nas quintas das 14:00 às 16:00 (sala CB14), exceto nas quintas-feiras 18/09, 02/10, 23/10, 13/11, 04/12 e 18/12. Nestes seis dias, haverá aula de laboratório na sala 302 do IC-3 (para mais detalhes ver a seção de [avaliação](#) nesta página);
- **atendimentos:** em horário a combinar (solicitar por *email* com pelo menos 48 horas de antecedência). Alternativamente, dúvidas poderão ser esclarecidas após o término das aulas. **Não haverá atendimento em semana de prova.**

● **Objetivos da disciplina:**

O objetivo desta disciplina é complementar a formação dos alunos na área de algoritmos, introduzindo técnicas para lidar com problemas *NP*-difíceis, resolvendo-os de forma heurística, exata ou aproximada. Espera-se que ao final do semestre os alunos tenham adquirido sólidos conhecimentos sobre as técnicas e sejam capazes de compreender as limitações de uso de cada uma delas.

● **Programa:** (em verde encontra-se o material já coberto em sala de aula)

**Algoritmos exatos para problemas NP-difíceis:** algoritmos pseudo-polinomiais para o problema da mochila; algoritmos de backtracking; algoritmos de branch-and-bound (exemplos: mochila binária e flowshop); **Programação Linear e Programação Linear Inteira:** formulando problemas usando PL; fluxos em redes via PL; resolução gráfica de um PL (no plano); descrição do algoritmo SIMPLEX: corretude (dualidade) e complexidade; Programação Linear Inteira como uma ferramenta para resolver problemas NP-difíceis; **Algoritmos heurísticos para problemas NP-difíceis:** definições básicas; algoritmos construtivos e algoritmos de busca local; algoritmos construtivos gulosos; algoritmos de busca local; meta-heurísticas; **Algoritmos aproximados** definições básicas; aproximação absoluta, fator de aproximação; exemplos de aproximação absoluta; inaproximabilidade em aproximação absoluta; exemplos de fator de aproximação; inaproximabilidade em fator de aproximação; esquemas de aproximação polinomial (definição e exemplo); uso de PL no desenvolvimento de algoritmos aproximados;

● **Referências bibliográficas:**

A seguir encontra-se a bibliografia de base da disciplina com alguns comentários adicionados pelo docente visando auxiliá-lo na escolha das obras a serem pesquisadas durante os seus estudos. Note que, além desses livros, existem nas bibliotecas da UNICAMP outras excelentes obras sobre os assuntos que serão vistos em sala de aula.

1. T.H. Cormen, C.E. Leiserson, R.L.Rivest e C. Stein. Introduction to Algorithms. 2nd Edition, McGraw-Hill, 2001.  
*Comentário:* Livro básico das disciplinas de algoritmos do IC.
2. U. Manber. Introduction to Algorithms: A Creative Approach. Addison-Wesley, 1989.  
*Comentário:* Outro livro básico das disciplinas de algoritmos do IC. O capítulo 10 trata exclusivamente sobre reduções entre problemas e bastante rico em exemplos. O mesmo pode ser dito a respeito do capítulo 11 sobre *NP*-completude.
3. C. H. Papadimitriou e K. Steiglitz. Combinatorial Optimization: Algorithms and Complexity. Prentice-Hall, Inc., 1982.  
*Comentário:* Uma boa fonte de referência na parte de *NP*-completude, principalmente para explicar as técnicas de *branch-and-bound* e apresentar os conceitos básicos de algoritmos aproximados.
4. E. Horowitz e S. Sahni. Fundamentals of Computer Algorithms. Computer Science Press, 1978.  
*Comentário:* Outra boa fonte de referência para *NP*-completude, algoritmos aproximados, algoritmos de *branch-and-bound* e *backtracking*.
5. M. Garey e D. Johnson. Computers and Intractability: a Guide to the Theory of NP-Completeness. Freeman, 1979.  
*Comentário:* Uma espécie de referência "bíblica" sobre a Teoria da Complexidade !
6. P. J. de Rezende e J. Stolfi. Fundamentos de geometria computacional. Universidade Federal de Pernambuco, Departamento de Informatica, 1994. IX Escola de Computação, Recife, 24 a 31 de julho de 1994.  
*Comentário:* Uma boa leitura sobre reduções entre problemas.
7. M.C. Goldberg e H.P.L. Luna. Otimização Combinatória e Programação Linear: modelos e algoritmos. Editora Campus, 2000.  
*Comentário:* Uma boa fonte de consulta em português sobre formulação de problemas usando PL e PLI.
8. M. Bazararaa, J. Jarvis e H. Sherali. Linear Programming and Network Flows. 2ª edição, John Wiley and Sons, 1990.  
*Comentário:* Uma boa fonte de consulta sobre PL e PLI.
9. L. Wolsey. Integer Programming. Wiley-Interscience, 1998.  
*Comentário:* Uma boa fonte de consulta sobre PLI.
10. M. H. Carvalho, M. R. Cerioli, R. Dahab, P. Feofiloff, C. G. Fernandes, C. E. Ferreira, K. S. Guimarães, F. K. Miyazawa, J. C. Pina Jr., J. Soares, Y. Wakabayashi, *Uma Introdução Sucinta a Algoritmos de Aproximação*, 23º Colóquio Brasileiro de Matemática, 2001 ([versão PDF](#), disponibilizada na página do [Prof. Flávio Miyazawa](#)).  
*Comentário:* Uma boa fonte de consulta sobre algoritmos aproximados.

● **Material didático:**

Podem ser baixados a seguir conjuntos de transparências que deverão ser usadas em sala de aula ao longo do semestre (formato PDF ou PS).

**Comentário:** Embora o conteúdo das transparências possa vir a ser alterado antes de uma aula, o conteúdo básico deverá permanecer o mesmo. Avisos sobre mudanças serão feitos em sala quando necessário.

- [Backtracking: conceitos gerais e aplicação ao problema SOS](#)
- [Backtracking: problema de coloração de grafos](#)  
Para estudar sobre Backtracking, recomenda-se fortemente que você leia a referência bibliográfica 4 (Horowitz e Sahni) pelo menos da página 323 a 369, as quais serviram de base para a preparação dos slides usados em aula. Note-se que, o estudo do restante do material no mesmo capítulo desta obra é igualmente relevante para o aprendizado do assunto.
- [Mochila Binária: Algoritmo de Programação Dinâmica](#)
- [Branch-and-bound: princípios gerais e exemplos \(mochila e flowshop\)](#)  
Para estudar sobre Branch-and-bound, recomenda-se fortemente que você leia a referência bibliográfica 3 (Papadimitriou e Steiglitz) pelo menos da página 438 a 448, as quais serviram de base para a preparação dos slides usados em aula. Note-se que, o estudo do restante do material no mesmo capítulo desta obra é igualmente relevante para o aprendizado do assunto.
- [Formulação de Problemas usando Programação Linear \(PL\)](#)  
*Este assunto pode ser estudado em qualquer bom livro de Programação Linear (veja, p.ex., a referência 8 nesta página).*
- [Fluxos em Redes e solução gráfica de PL \(no plano\)](#)  
*Este assunto pode ser estudado em qualquer bom livro de Programação Linear (veja, p.ex., a referência 8 nesta página).*
- [Resolução de sistemas lineares e o algoritmo SIMPLEX](#)
- [Dualidade em PL](#)
- [Regra de Bland \(evita ciclagem do SIMPLEX\) e a Fase I](#)
- [Complexidade do Simplex.](#)
- [Formulação PL I](#)
- [Formulação PL II + TU](#)
- [Heurísticas construtivas e busca local](#)
- [Heurísticas construtivas e busca local](#) (gentilmente cedidas pelo [Prof. Flávio Miyazawa](#))
- [Metaheurísticas](#) (gentilmente cedidas pelo [Prof. Flávio Miyazawa](#))
- [GRASP](#) (gentilmente cedidas pelo [Prof. Flávio Miyazawa](#))
- [Heurísticas+algoritmos aproximados](#) (material preparado pelo docente da disciplina)  
Neste último conjunto de transparências, o material sobre algoritmos aproximados teve por base o conteúdo das páginas 559-577 e 363-367 das referências 4 e 2 listadas [abaixo](#), respectivamente.
- [Prog. Linear e Algoritmos de Aproximação.](#)
- [Algoritmo de Graham \(2-aproximação\) para um problema de escalonamento](#)
- [1.5-aproximação para o TSP métrico](#) (gentilmente cedidas pelo [Prof. Flávio Miyazawa](#))
- [Esquemas de Aproximação Completamente Polinomiais \(FPTAS\)](#) (material preparado pelo docente da disciplina)

#### ● Avaliação:

A avaliação será baseada nas notas de duas provas, denotadas por P1 e P2, e nas notas de seis trabalhos práticos de implementação, representadas por T1, T2, T3, T4, T5 e T6. Todos trabalhos práticos serão feitos total ou parcialmente nas aulas de laboratório.

A partir destas avaliações, e sendo F a frequência do aluno nas aulas de laboratório, medida sob a forma de uma fração, a sua nota será calculada da seguinte forma:

- **Média dos laboratórios:**  $MT = F \times ((T1+T2+T3+T4+T5+T6)/6)$
- **Média das provas:**  
se  $P1 \geq 4.0$  e  $P2 \geq 4.0$  então  $MP = (P1+P2)/2$   
se não  $MP = \min \{ 4.9, (P1+P2)/2 \}$
- **Média do semestre:** (antes do exame)  
se  $MP \geq 5.0$  e  $MT \geq 5.0$  então  $MS = (3 \times MP + MT)/4$   
se não  $MS = \min \{ 4.9, (3 \times MP + MT)/4 \}$   
**Observação:** se  $MS < 2.5$ , o aluno está impedido de fazer o exame
- **Média final:** (após o exame)  
se o aluno fez o exame então  $MF = \min \{ 5.0, (MS+E)/2 \}$   
se não
  - se  $MS < 2.5$  ou  $MS \geq 5.0$  então  $MF = MS$
  - se não  $MF = MS/2$
- **Resultado final:**  
se o aluno não teve a frequência mínima então ele **REPROVOU-SE** por falta  
se não
  - se  $MF \geq 5.0$  então o aluno **APROVOU-SE**
  - se não o aluno **REPROVOU-SE** por nota

#### Observações:

1. Durante o semestre não serão ministradas provas ou trabalhos práticos antecipados nem substitutivos.
2. Todas as provas realizadas durante o semestre e o exame final serão sem consulta.
3. Qualquer tentativa de fraude nas provas ou nos trabalhos práticos implicará em média do semestre (MS) igual a ZERO para todos os envolvidos, sem prejuízo de outras sanções.
4. Qualquer tentativa de fraude no exame implicará em média final (MF) igual a ZERO para todos os envolvidos, sem prejuízo de outras sanções.
5. Qualquer **pedido de revisão** de uma nota divulgada durante o semestre, seja prova ou laboratório, deverá ser feito dentro de um prazo **máximo de 7 dias** (corridos) contados a partir da data de divulgação do resultado correspondente. No caso do **exame**, a revisão terá que ser feita no dia seguinte à realização da prova em horário determinado pelo docente.

● **Listas de exercícios:** seguem abaixo algumas listas de exercícios sobre tópicos que será cobertos neste semestre. Outras listas poderão ser propostas mas nenhuma delas será corrigida.

1. [Listas de exercícios de modelagem usando PL](#)
2. [Backtracking, Branch-and-bound, PL I](#)  
Outras listas de semestres anteriores sobre este mesmos assuntos:
  - [Backtracking e Branch-and-bound](#) (Prof. Eduardo Xavier),
  - [Formulação PL I](#) (Prof. Eduardo Xavier),
  - [Backtracking, Branch-and-bound, PL I](#) (Prof. Flávio Miyazawa).
3. [Dualidade em PL](#)
4. [Algoritmos Aproximados](#) (preparada pelo docente)
5. [Algoritmos Aproximados](#) (disponibilizada pelo [Prof. Eduardo Xavier](#))

#### ● **Datas Importantes:**

- [Calendário oficial da DAC](#). Visite esta página para saber quais as datas de alteração de matrícula, desistência de disciplinas e dos períodos sem atividade.
- 02/09 (ter): início das aulas.

- 18/09 (qui): 1º laboratório.
  - 02/10 (qui): 2º laboratório.
  - 21/10 (ter): 1ª prova (P1).
  - 23/10 (qui): 3º laboratório.
  - 13/11 (qui): 4º laboratório.
  - 04/12 (qui): 5º laboratório.
  - 16/12 (ter): 2ª prova (P2).
  - 18/12 (qui): 6º laboratório.
  - 15/01 (qui): exame (E).
  - 16/01 (qui): revisão de notas do exame, à tarde (**único dia !**)
-