Iterated Local Search (ILS)

Marcone Jamilson Freitas Souza[1,2,3]

Puca Huachi Vaz Penna[1]

[1] Departamento de Computação

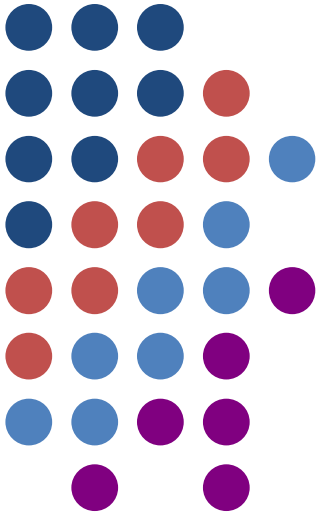
[1] Programa de Pós-Graduação em Ciência da Computação

Universidade Federal de Ouro Preto

[2] Programa de Pós-graduação em Modelagem Matemática e Computacional / CEFET-MG

[3] Programa de Pós-graduação em Instrumentação, Controle e Automação de Processos de Mineração / ITV/UFOP

www.decom.ufop.br/prof/marcone, www.decom.ufop.br/puca
E-mail: {marcone,puca}@ufop.edu.br



Souza, Marcone J. F. e Penna, Puca H. V. Iterated Local Search. Notas de aula de Técnicas Metaheurísticas para Otimização Combinatória. Departamento de Computação, Universidade Federal de Ouro Preto, Ouro Preto, 2021. Disponível em www.decom.ufop.br/prof/marcone/Disciplinas/InteligenciaComputacional/ILS.pptx

Histórico do ILS

- Algoritmos precursores:
  - Baxter, J. (1981). Local optima avoidance in depot location. Journal of the Operational Research Society, 32(9):815–819.
  - E.B. Baum, Iterated descent: A better algorithm for local search in combinatorial optimization problems, Technical report, Caltech, Pasadena, CA, 1986.
  - D. S. Johnson. Local optimization and the travelling salesman problem. In Proceedings of the 17th Colloquium on Automata, Languages, and Programming, volume 443 of LNCS, pages 446–461. Springer Verlag, Berlin, 1990.
  - O. Martin, S.W. Otto, E.W. Felten. Large-step Markov chains for the traveling salesman problem, Complex Systems, 5:299–326, 1991.
  - H. R. Lourenço. Job-shop scheduling: Computational study of local search and large-step optimization methods. European Journal of Operational Research, 83:347–364, 1995.
  - O. Martin, S.W. Otto. Combining simulated annealing with local search heuristics, Annals of Operations Research, 63:57-75, 1996.
  - N. Mladenovic and P. Hansen. Variable Neighborhood Search. Computers & Operations Research, 24:1097–1100, 1997.

Histórico do ILS

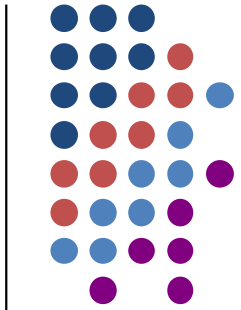
- "Redescoberto" e nomeado *Iterated Local Search* em:
  - Thomas Stützle. Local Search Algorithms for Combinatorial Problems Analysis, Improvements, and New Applications. PhD thesis, Darmstadt University of Technology, Department of Computer Science, Darmstadt, Germany, 1998.
- Artigos mais citados:
  - H.R. Lourenço, O. Martin, T. Stützle. A Beginner's Introduction to Iterated Local Search. Proceedings of the 4th Metaheuristic International Conference, Porto, Portugal, 2001.
  - H.R. Lourenço, O. Martin, T. Stützle. Iterated local search, in: F. Glover, G. Kochenberger (Eds.), Handbook of Metaheuristics, International Series in Operations Research & Management Science, vol. 57, Kluwer Academic Publishers, Norwell, MA, 2002, pp. 321–353.

Fundamentação do ILS

- Método que consiste em explorar o espaço de soluções por meio de perturbações em ótimos locais
- Pressuposto:
  - Os ótimos locais de um problema de otimização podem ser gerados a partir de perturbações em uma solução ótima local corrente
- A perturbação precisa ser suficientemente forte para permitir que a busca local explore diferentes soluções e fraca o suficiente para evitar um reinício aleatório

Componentes do ILS

- GeraSolucaoInicial:
  - Produz uma solução inicial
- BuscaLocal:
  - Retorna uma solução melhorada
- Perturbacao:
  - Modifica a solução corrente guiando a uma solução intermediária
- CriterioAceitacao:
  - Decide de qual solução a próxima perturbação será aplicada



Algoritmo

Algoritmo ILS

$s_0 \leftarrow$ *SolucaoInicial*

$s \leftarrow$ *BuscaLocal*(s_0)

iter ← 0; {Contador do número de iterações}

MelhorIter ← Iter; {Iteração em que ocorreu melhora}

enquanto (*iter* – MelhorIter < ILS_{max})

iter ← *iter* + 1

s' ← *perturbação*(*s*, *histórico*)

s" ← *BuscaLocal*(*s'*)

s ← *CriterioAceitacao*(*s*, *s"*)

fim-enquanto

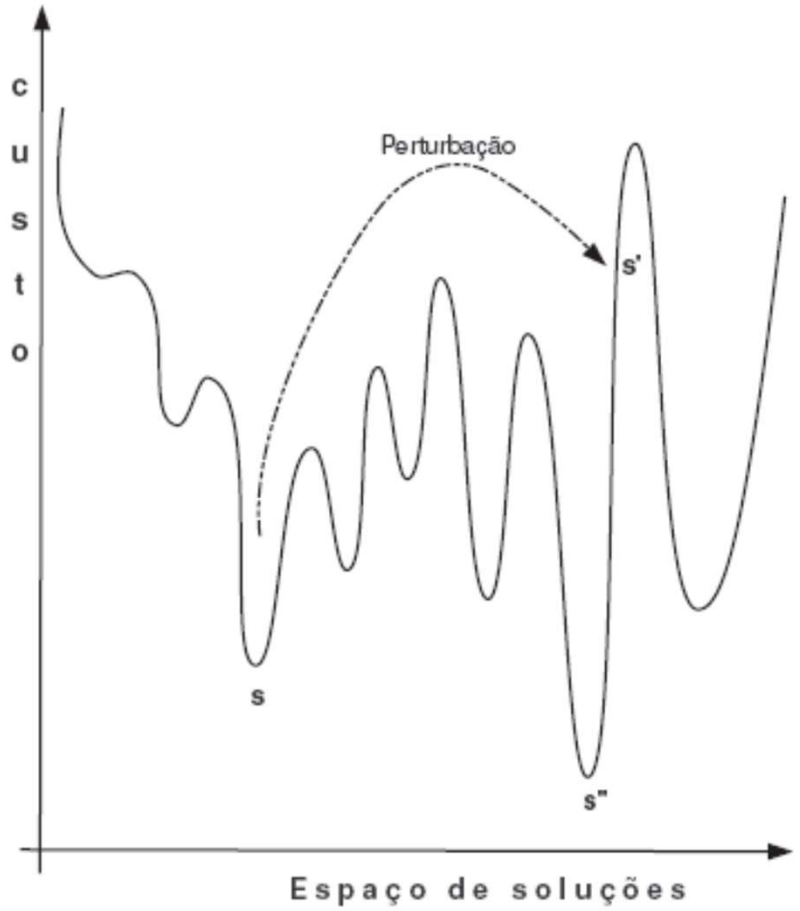
retorne *s*

se (*f*(*s"*) < *f*(*s*)) **faça**

s ← *s"*

fim-se

Ilustração do funcionamento do ILS



Intensificação x Diversificação

- Combina intensificação com diversificação
- Intensificação:
  - Consiste em explorar a região atual de busca
  - É obtida fazendo-se “pequenas” perturbações na solução ótima local corrente
- Diversificação:
  - Consiste em mudar a região de busca
  - É obtida aumentando-se gradativamente a quantidade de perturbações na solução ótima local corrente

Descrição do método para um problema de minimização

$s_0 \leftarrow$ SolucaoInicial();
s $\leftarrow$ BuscaLocal(s_0);
iter $\leftarrow$ 0; MelhorIter $\leftarrow$ Iter; nivel $\leftarrow$ 1;
enquanto (iter - melhorIter < ILSmax) **faça**
    iter $\leftarrow$ iter + 1;
    s' $\leftarrow$ perturbacao(s, nivel);
    s'' $\leftarrow$ BuscaLocal(s');
    se (f(s'') < f(s)) **então**
        s $\leftarrow$ s'';
        melhorIter $\leftarrow$ iter;
        nivel $\leftarrow$ 1;
    senão
        nivel $\leftarrow$ nivel + 1;
    fim-se
fim-enquanto

Descrição do método de perturbação

```
procedimento perturbação(s, nível)
  s' ← s;
  nmodificacoes ← nivel + 1;
  cont ← 1;
  enquanto ( cont ≤ nmodificacoes ) faça
     Aplique movimento aleatório em s';
     cont ← cont + 1;
  fim-enquanto
retorne s'
```

Smart ILS

- Ideia básica:
  - Aumentar o nível de perturbação somente após algumas tentativas sem sucesso
  - Justificativa: a região de busca pode não ter sido explorada adequadamente
- Usado em diversos trabalhos, entre eles:
  - Coelho, V.N.; Grasas, A.; Ramalhinho, H.; Coelho, I.M.; Souza, M.J.F.; Cruz, R.C. An ILS-based algorithm to solve a large-scale real heterogeneous fleet VRP with multi-trips and docking constraints. European Journal of Operational Research, 250: 367-376, 2016.
- Nomeado *Smart ILS* em:
  - Reinsma, J. A.; Penna, P. H. V.; Souza, M. J. F. Um algoritmo simples e eficiente para resolução do problema do caixeiro viajante generalizado. Anais do 50º Simpósio Brasileiro de Pesquisa Operacional. Rio de Janeiro: SOBRAPO, 2018.

Smart ILS

$s_0 \leftarrow$ SolucaoInicial();
s $\leftarrow$ BuscaLocal(s_0);
iter $\leftarrow$ 0; MelhorIter $\leftarrow$ Iter; nivel $\leftarrow$ 1; nvezes $\leftarrow$ 1;
enquanto (iter - melhorIter < ILSmax) **faça**
    iter $\leftarrow$ iter + 1;
    s' $\leftarrow$ perturbacao(s, nivel);
    s'' $\leftarrow$ BuscaLocal(s');
    se (f(s'') < f(s)) **então**
        s $\leftarrow$ s''; melhorIter $\leftarrow$ iter; nivel $\leftarrow$ 1; nvezes $\leftarrow$ 1;
    senão
        se (nvezes ≥ vezesMax) **então**
            nivel $\leftarrow$ nivel + 1; nvezes $\leftarrow$ 1;
        senão
            nvezes $\leftarrow$ nvezes + 1;
        fim-se
    fim-se
fim-enquanto