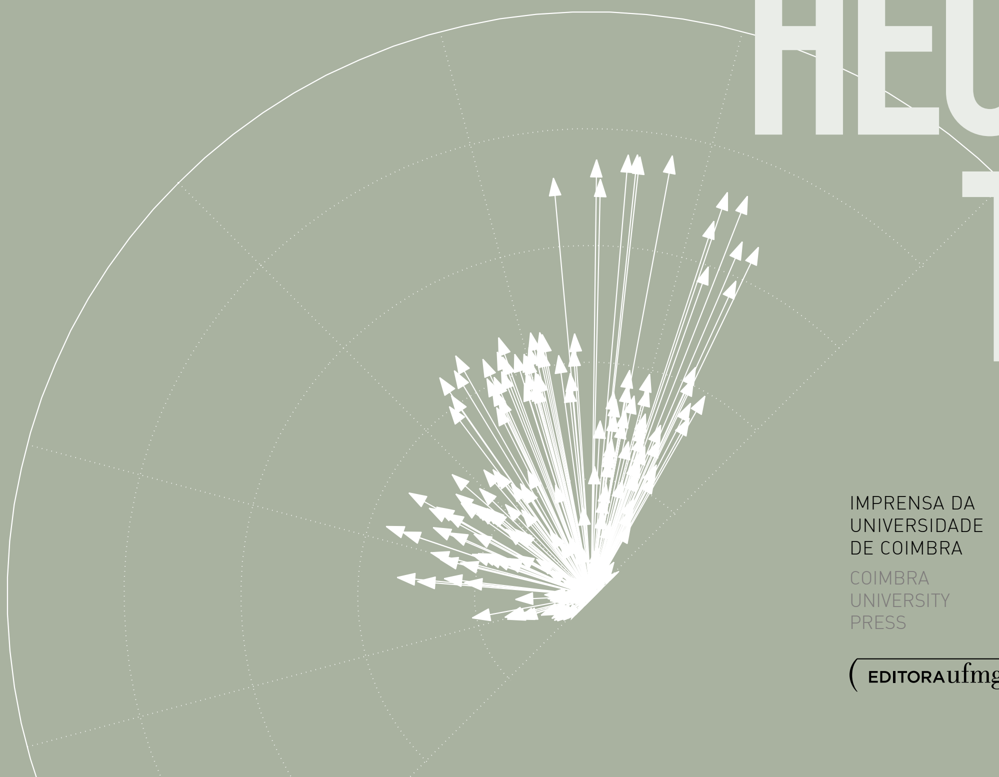
MANUAL DE COMPUTAÇÃO EVOLUTIVA E META HEURÍS TICA

ANTÓNIO GASPAR-CUNHA
RICARDO TAKAHASHI
CARLOS HENGGELER ANTUNES
COORDENADORES



IMPRENSA DA UNIVERSIDADE DE COIMBRA
COIMBRA UNIVERSITY PRESS

EDITORA UFMG

CAPÍTULO 12

Pesquisa Local Iterativa e em Vizinhança Variável

Luís Paquete

Departamento de Engenharia Informática
Universidade de Coimbra

As metaheurísticas têm tido grande sucesso na resolução de imensos problemas de optimização que surgem tanto a nível académico como a nível industrial. Contudo, o desenho de metaheurísticas não é um exercício trivial. Para obter um desempenho competitivo com metaheurísticas é necessário ter um conhecimento muito aprofundado do estado-da-arte dos métodos existentes e do próprio problema em estudo.

Nas várias metaheurísticas, como a pesquisa local iterativa (Lourenço et al., 2002), pesquisa tabu (Glover, 1989) e pesquisa em vizinhança variável (Mladenović e Hansen, 1997), assume-se que existe uma *heurística* que toma decisões locais com base no conhecimento do problema.[1] O modo de funcionamento desta classe de metaheurísticas é "guiar" essa heurística no espaço de pesquisa com base na escolha apropriada de determinados parâmetros e métodos que resultam muitas vezes de experimentação algoritmica. A clara vantagem destas abordagens é que são conceptualmente simples de parameterizar e bastante eficazes, uma vez encontrada uma boa heurística para o problema em estudo.

Tanto a pesquisa local iterativa (ILS) como a pesquisa de vizinhança variável (VNS)[2] são duas das

---

[1] É possível encontrar o termo "Procura" e "Busca" em vez de "Pesquisa" em trabalhos científicos publicados em Portugal e no Brasil, respectivamente.

[2] Neste capítulo utilizaremos as iniciais da denominação anglo-saxónica, respectivamente, ILS (*Iterated Local Search*) e VNS (*Variable Neighborhood Search*).

metaheurísticas mais simples de parameterizar, sem que isso signifique uma perda de desempenho. Em particular, ILS tem demonstrado um desempenho bastante elevado em vários problemas clássicos nas Ciências da Computação e Investigação Operacional, tal como o problema do caixeiro viajante (Martin et al., 1991b; Codenotti et al., 1996; Johnson e McGeoch, 1997; Stützle e Hoos, 2002), problemas de escalonamento (Lourenço e Zwijnenburg, 1996; Balas e Vazacopoulos, 1998; Yang et al., 2000), o problema de afectação quadrática (Stützle, 2006) e o problema de coloração de grafos (Paquete e Stützle, 2002).

O *modus operandis* da ILS consiste em construir, de uma forma iterativa, uma sequência de soluções geradas por uma heurística subordinada. Em cada iteração, a solução inicial da heurística é obtida através de uma *perturbação* na solução retornada por essa heurística numa iteração anterior. Nas primeiras abordagens de ILS, a heurística consistia de um método estocástico de pesquisa local que retornava unicamente uma solução. Contudo, é possível encontrar variantes mais recentes de ILS em que a heurística subordinada é determinística, constructiva e/ou produz uma população de soluções tal como os algoritmos evolutivos. A ILS foi proposta por vários investigadores de forma independente, sendo conhecida por *large-step Markov chains* (Martin et al., 1991b) ou *iterated Lin-Kernighan* (Johnson e McGeoch, 1997). Recentemente, um grupo de investigadores considerou que estas abordagens seguiam princípios muito semelhantes, daí resultando a denominação de ILS numa tentativa de unificação (Lourenço et al., 2002).

A VNS segue um princípio muito semelhante ao da ILS. Ambas executam uma sequência de heurísticas subordinadas, alternadas com pequenas perturbações nas soluções retornadas por essas heurísticas. Por esta razão, a VNS é considerada como um caso particular de ILS. A ideia foi proposta por Mladenović e Hansen (1997) e embora não tenha ainda um desempenho tão marcante como ILS para o problema do caixeiro viajante, o conceito é bastante simples de implementar e tem apresentado resultados bastante promissores em vários problemas combinatórios, como para o problema de *p-mediana* (Hansen e Mladenović, 1997), problemas de optimização em *clustering* (Hansen e Mladenović, 2001) e de satisfiabilidade (Hansen et al., 2000).

O objectivo deste capítulo é apresentar uma descrição detalhada dos princípios da ILS e VNS e explicar casos de sucesso da sua aplicação em problemas de optimização. Assume-se que o leitor tenha conhecimentos básicos de optimização combinatória e métodos de pesquisa local (ver, p.e. Papadimitriou e Steiglitz (1982) e Hoos e Stützle (2004)). O capítulo está organizado da seguinte forma. As Secções 1 e 2 descrevem os fundamentos de ILS e os componentes algorítmicos desta metaheurística, respectivamente. Seguidamente, a Secção 3 apresenta um caso de estudo. A VNS é apresentada na Secção 4. Finalmente, temas para investigação futura nesta área são apresentados na Secção 5.

1. Fundamentos de Pesquisa Local Iterativa

Esta secção descreve os princípios básicos de ILS, tal como descritos em (Lourenço et al., 2002). Começamos por introduzir alguns conceitos introdutórios que permitirão uma compreensão mais alargada do funcionamento desta metaheurística.

Dado um determinado problema de optimização discreta , assume-se que existe uma heurística subordinada que devolve uma solução admissível para esse problema. Iremos denominar essa heurística de `PesquisaLocal`, visto que toma decisões a nível local.

Para um dado problema P com uma função objectivo f definimos como S o conjunto finito de soluções admissíveis desse problema. Sem perda de generalidade, assumimos que o objectivo do problema P é encontrar uma solução em S que minimize a função f. Para manter a simplicidade da exposição, assumimos igualmente que `PesquisaLocal` apresenta as seguintes especificidades: (i) é um método de pesquisa local; (ii) inicia o processo de pesquisa numa solução escolhida aleatoriamente em S; (iii) em cada iteração escolhe a solução vizinha com menor valor da função f (menor custo) e; (iv) termina num óptimo local.

Associada ao método de pesquisa local e ao problema P existe uma noção de vizinhança N que, quando *adicionada* ao conjunto S, forma uma determinada estrutura topológica $\mathcal{S}_N$, também conhecida por *espaço de pesquisa*. Em problemas discretos, esta estrutura pode ser representada por um grafo dirigido em que cada solução admissível é um nó e cada arco liga um nó u a outro nó v se e só se u é vizinho de v de acordo com a vizinhança N. Se considerarmos a informação da função f, este grafo passa a ter ponderações nos arcos que indicam a diferença de custo entre cada par de nós vizinhos, isto é, o peso do arco que liga o nó u ao nó v corresponde a $f(v) - f(u)$. A heurística `PesquisaLocal` pode ser entendida como um algoritmo que atravessa o grafo $\mathcal{S}_N$ a partir de um determinado nó inicial. Este algoritmo escolhe, iterativamente, o nó vizinho cujo o arco que o liga ao nó actual tem o menor peso e termina quando encontra um nó $s^\ell \in S_N$ com arcos com pesos não-negativos, isto é, s^ℓ é um óptimo local. Com base nesta representação do problema de pesquisa local podemos afirmar que a heurística `PesquisaLocal` mapeia cada solução admissível em S num elemento do conjunto $S^\ell \subseteq S$ de óptimos locais. Realçamos que a solução que minimiza a função objectivo também é um óptimo local, mas que está muito pouco acessível a `PesquisaLocal` devido às características do grafo S_N.

A distribuição do custo dos elementos em S e dos elementos em S^ℓ são tipicamente diferentes para o mesmo problema. É de esperar que esta última seja caracterizada por uma média e variância significativamente inferiores à média e variância do custo dos elementos em S. Por esta razão, será preferível efectuar uma amostragem em S^ℓ do que em S. Obviamente, esta observação só será válida se, na prática, o tempo computacional para obter uma solução em S^ℓ, por exemplo através de `PesquisaLocal`, for suficientemente curto. Por esta razão, é particularmente relevante considerar métodos incrementais de avaliação de soluções vizinhas de uma forma eficiente. De facto, existem muitas formas de o fazer em problemas discretos. Infelizmente, vários investigadores têm demonstrado que o número de vizinhos a visitar até encontrar um óptimo local é exponencial relativamente ao tamanho do problema no cenário de pior caso (Johnson et al., 1988). Contudo, outros investigadores têm observado que este tempo é negligível na prática, o que indica que a complexidade do problema de encontrar um óptimo local seja limitada por uma função polinomial no caso médio, tal como no caso do algoritmo de Simplex para problemas de programação linear (Klee e Minty, 1972).

Dada a vantagem em efectuar uma amostragem de S^ℓ, a principal questão que se coloca agora é como efectuar esse procedimento. O método mais simples consiste em executar `PesquisaLocal` múltiplas vezes a partir de uma solução escolhida aleatoriamente em S. Logo, cada óptimo local gerado por `PesquisaLocal` em cada uma dessas execuções é independente. Neste caso, podemos considerar que `PesquisaLocal` está a efectuar uma amostragem independente em S^ℓ. Contudo, a média amostral do custo dos óptimos locais está tipicamente localizada a uma percentagem fixa do custo do óptimo global (Lourenço et al., 2002). Por esta razão, supõe-se que este processo de amostragem tem uma probabilidade cada vez menor de encontrar bons óptimos locais com o aumento do tamanho da instância. Uma possibilidade de ultrapassar esta desvantagem é recorrer a um processo de amostragem enviesada, tirando partido de alguma estrutura do problema.

É possível idealizar um método de pesquisa local que, de acordo com uma determinada noção de vizinhança N^ℓ, pesquisa unicamente em S^ℓ até encontrar um óptimo local relativamente a N^ℓ. Espera-se que a distribuição do custo desses óptimos locais de acordo com a vizinhança N^ℓ tenha uma média e variância inferiores à média e à variância da distribuição dos custos dos óptimos locais em S^ℓ. Claramente, este princípio não é possível de ser implementado na prática porque tal noção de vizinhança não é conhecida *a priori*. Contudo, podemos definir o que seria desejável nessa vizinhança N^ℓ: Dado dois conjuntos de soluções S^1 e S^2 que pertencem a duas bacias de atracção distintas[3], estes dois conjuntos são vizinhos em N^ℓ se existe pelo menos uma solução $s^1 \in S^1$ e outra solução $s^2 \in S^2$ e s^1 e s^2 são vizinhas relativamente a N. Por outras palavras, as bacias de atracção são vizinhas se ambas se *tocam*. Claramente, o ideal é conceber um algoritmo que efectue uma caminhada em $\mathcal{S}_N$ que

[3] Para qualquer solução inicial numa bacia de atracção, o algoritmo `PesquisaLocal` irá sempre encontrar o mesmo óptimo local.

Algoritmo 1 Pesquisa Local Iterativa

$s \leftarrow$ `Gera`()
$s^\ell \leftarrow$ `PesquisaLocal`(s)
repita
    $s \leftarrow$ `Perturba`(s^ℓ, memória)
    $s^{\ell *} \leftarrow$ `PesquisaLocal`(s)
    $s^\ell \leftarrow$ `Aceita`($s^\ell, s^{\ell *}$, memória)
até condição de paragem ser verdadeira

permita sair de um bacia de atracção e entrar noutra bacia de atracção vizinha. Contudo, sabemos que é impossível encontrar esse caminho de um modo determinístico em tempo razoável, assim como saber a que bacia de atracção é que uma determinada solução pertence.

ILS tenta efectuar uma caminhada entre duas bacias de atracção de uma forma estocástica e heurística em cada iteração. A ideia é perturbar um óptimo local $s^\ell \in S^\ell$ para gerar uma solução $s \in S$; seguidamente, `PesquisaLocal` inicia a pesquisa a partir de s para retornar outro óptimo local $s^{\ell *} \in S^\ell$; finalmente, um determinado critério de aceitação decide qual dos óptimos locais encontrados, s^ℓ ou $s^{\ell *}$, é que vai ser perturbado na iteração seguinte. Claramente, ILS efectua uma pesquisa em S^ℓ mas sem utilizar uma noção explícita de vizinhança entre bacias de atracção.

O Algoritmo 1 apresenta o pseudo-código de ILS. Para evitar que ILS entre facilmente em ciclos durante a pesquisa em S^ℓ, consideramos "memória" como uma estrutura de dados adicional que retorna informação sobre as soluções já visitadas. Esta estrutura está associado à perturbação (`Perturba`) e ao critério de aceitação (`Aceita`). Estudos recentes indicam que a presença de memória melhora o desempenho de ILS. Realçamos que a definição da perturbação é naturalmente dependente do problema, mas o critério de aceitação pode ser definido de uma forma independente. Por exemplo, o critério de aceitação pode escolher a solução com o menor custo, tal como é efectuado pelo método de pesquisa local. É possível também considerar outros critérios de aceitação baseados em arrefecimento simulado ou pesquisa tabu.

2. Componentes de Pesquisa Local Iterativa

Uma grande vantagem do ILS, de um ponto de vista mais pragmático, é a sua modularidade. Como pode-se observar no Algoritmo 1, é possível instanciar várias configurações de ILS ao modificar os componentes algorítmicos `Gera`, `PesquisaLocal`, `Perturba` e `Aceita`. A optimização do ILS consiste em encontrar uma combinação destes quatro componentes que maximize o seu desempenho. Naturalmente, o investigador pode optimizar cada um destes componentes de um modo sequencial, sem considerar as interações entre estes, o que produz uma configuração sub-óptima de ILS. Realçamos que o problema de encontrar uma configuração óptima de uma metaheurística, quer seja ILS, algoritmos evolutivos ou colónia de formigas, é ainda uma questão em aberto. Contudo, os resultados experimentais obtidos por um grande número de investigadores indicam que uma boa afinação de metaheurísticas está dependente da experiência do investigador na abordagem que escolheu e no conhecimento que este tem do problema em estudo. Actualmente existem vários métodos que afinam metaheurísticas de uma forma automática com base em inferência estatística (ver Birattari et al. (2002)). Contudo, mesmo nestes métodos automáticos é necessário conhecer os efeitos de cada componente no desempenho de uma metaheurística. No texto que se segue apresentamos algumas sugestões que podem apoiar o investigador a efectuar as primeiras escolhas algorítmicas para um dado problema.

Geração de Solução Inicial O componente `Gera` é responsável pela geração da solução inicial da ILS. Existem duas maneiras conhecidas de gerar esta solução: aleatoriamente ou com base numa *heurística gulosa*. Vários investigadores consideram que a segunda opção é preferível, principalmente porque a solução retornada pela heurística gulosa já apresenta algum nível de qualidade e porque estas heurísticas são muito eficientes do ponto de vista de tempo computacional. Contudo, é necessário ter algum cuidado com a escolha da heurística gulosa, pois esta pode retornar soluções que não são possíveis de serem melhoradas através do componente `PesquisaLocal`. Por exemplo, os resultados experimentais de Johnson e McGeoch (1997) no problema do caixeiro viajante indicam que o método de pesquisa local utilizado por estes autores apresenta melhor desempenho quando a solução inicial é gerada aleatoriamente do que através da heurística de Clarke-Wright. Por outro lado, tem-se verificado que existe pouca relação entre a qualidade da solução inicial e a qualidade da solução final quando se consideram execuções muito longas da ILS (Lourenço et al., 2002).

Perturbação A perturbação é normalmente definida por um parâmetro a que chamamos *força*. Por exemplo, se a perturbação efectua k iterações de uma pesquisa local que selecciona vizinhos de uma forma aleatória, então k é a força desta perturbação. A escolha de uma perturbação está claramente relacionada com a escolha da heurística para `PesquisaLocal`. Um requisito essencial é escolher uma perturbação que impeça a heurística `PesquisaLocal` de retornar uma solução numa bacia de atracção já visitada. Contudo, as perturbações não podem ser demasiado *fortes*, caso contrário, ILS não será nada mais do que um algoritmo de reinicialização aleatória. Normalmente, é necessário efectuar uma análise experimental com alguma profundidade para encontrar boas perturbações e/ou valores aceitáveis para o parâmetro de força. Resultados experimentais indicam que é preferível utilizar uma força pequena para o problema do caixeiro viajante, tal como a perturbação *double-bridge* (ver Secção 3), e para o problema de escalonamento de “flow shop” (Lourenço et al., 2002). Contudo, para o problema de afectação quadrática, a perturbação tem de alterar aproximadante 75% dos componentes da solução (Stützle, 2006).

Infelizmente, não é sempre possível obter uma perturbação simples e eficaz para muitos problemas. É possível utilizar um tipo de perturbação mais complexa em que, por exemplo, a força varie de acordo com o tempo computacional dispendido. Estes princípios adaptativos aproximam ILS de outras abordagens como pesquisa em vizinhança variável (Mladenović e Hansen, 1997) ou pesquisa reactiva (Battiti e Tecchiolli, 1994). Outra alternativa é considerar uma perturbação nos dados de entrada do problema e executar `PesquisaLocal` no problema modificado, tal como sugerido por Codenotti et al. (1996) para o problema do caixeiro viajante. Outra possibilidade é efectuar uma perturbação que resolve um subproblema, quer através de heurísticas ou mesmo de uma forma exacta, como proposto por Lourenço e Zwijnenburg (1996) para um problema de escalonamento.

Critério de Aceitação O componente `Aceita` determina qual dos dois óptimos locais, o actual ou aquele que foi retornado por `PesquisaLocal` numa iteração anterior, é escolhido para ser perturbado. Este procedimento permite ao investigador obter um balanço entre *exploração* e *intensificação* do ILS. Por exemplo, um critério de aceitação que aceita o melhor óptimo local para perturbação favorece claramente a intensificação. Por outro lado, um critério de aceitação que aceita sempre o óptimo local mais recente favorece a exploração do espaço de pesquisa.

Existem outras alternativas aos dois critérios de aceitação acima descritos. Por exemplo, é possível considerar um critério de aceitação baseado no algoritmo de arrefecimento simulado (Martin et al., 1991b). Neste caso existe uma probabilidade de aceitar um óptimo local com qualidade inferior de acordo com um parâmetro de temperatura. Este componente também pode ser combinado com alguma estrutura de dados adicional que mantém a memória dos passos anteriores efectuados por ILS. Por exemplo, se o critério de aceitação continua a aceitar o mesmo óptimo local por um determinado número de iterações, então fará sentido recorrer ao procedimento `Gera` para gerar outra solução para reinicializar ILS. Realçamos que, em qualquer caso, é necessário guardar a melhor solução encontrada até ao momento, pois essa é a solução que deverá ser retornada pelo ILS.

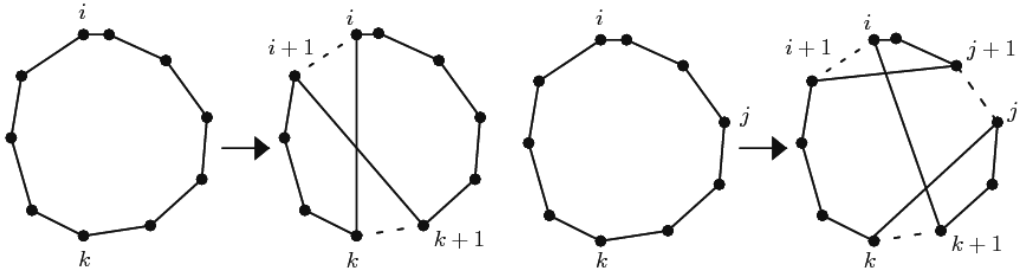


Figura 12.1: Ilustração da modificação do circuito com base no 2-opt e no 3-opt.

Pesquisa Local Para cada problema e para cada representação computacional desse problema podem existir diversas noções de vizinhança. Para cada uma dessas vizinhanças podem ainda existir diversas maneiras de a explorar, quer seja de um modo exaustivo, ou de um modo aleatório[4]. Finalmente, podem também existir diversas variantes para o critério de aceitação, semelhantes às descritas na secção anterior para o ILS. Infelizmente, não é possível determinar *a priori* qual é a melhor escolha para um dado problema e respectiva representação. Por isso, é necessário recorrer muitas vezes à experimentação. Espera-se que uma boa escolha para um método de pesquisa local seja também a melhor escolha para o componente `PesquisaLocal` do ILS. Contudo, é necessário ter em conta o tempo computacional disponível para executar um ILS, pois pode ser preferível ter um método de pesquisa local que retorne um óptimo local de inferior qualidade em muito pouco tempo do que um método que retorne um óptimo local de boa qualidade mas que requer bastante tempo computacional; no primeiro caso, ILS efectua mais iterações e pode explorar o espaço de pesquisa de um modo mais eficaz.

Um aspecto bastante importante no método de pesquisa local é a utilização de estruturas de dados e algoritmos que permitem a computação eficiente do custo das soluções vizinhas. Quanto mais eficiente é o método de pesquisa local, mais iterações de ILS é possível efectuar. Também não estamos restritos a métodos de pesquisa local no componente `PesquisaLocal`, pois é possível utilizar uma metaheurística como arrefecimento simulado ou mesmo pesquisa tabu (Lourenço e Zwijnenburg, 1996).

3. Um caso de estudo - O Problema do Caixeiro Viajante

O problema do caixeiro viajante é um problema clássico nas Ciências da Computação e na Investigação Operacional. Curiosamente, é neste problema que a metaheurística ILS apresenta o seu melhor desempenho. Na sua forma mais simples, este problema consiste em encontrar o circuito mais curto num conjunto de cidades em que cada cidade só pode ser visitada uma só vez. Os dados de entrada do problema podem ser representados por um grafo em que as cidades correspondem aos nós do grafo, os caminhos entre cidades correspondem às arestas do grafo e a distância entre cidades é o peso associado às arestas. O problema consiste em encontrar o ciclo Hamiltoniano mais curto nesse grafo.

Existem vários métodos de pesquisa local para este problema, mas os mais utilizados na literatura são o 2-opt e o 3-opt. O 2-opt baseia-se na troca de duas arestas não-adjacentes no circuito por

[4] Note que se o método de pesquisa local só considera a vizinhança de um modo parcial, então não é possível garantir que esse método retorna um óptimo local; contudo, como poderemos observar no caso de estudo da Secção 3, este pormenor é muitas vezes deixado de parte por questões de eficiência.

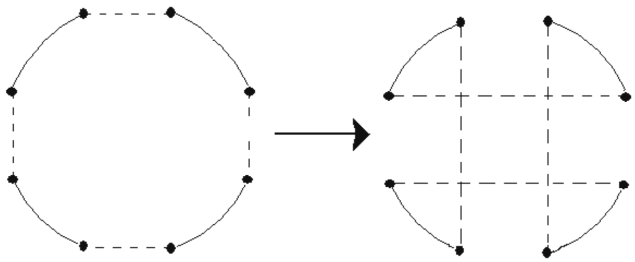


Figura 12.2: Ilustração da perturbação *double-bridge*.

outras duas arestas não-adjacentes que não estão presentes no circuito. O 3-opt é semelhante mas considera três arestas. A Figura 12.1 ilustra as modificações efectuadas no circuito. Um método de pesquisa mais complexo mas com melhor desempenho em termos da qualidade da solução retornada é o Lin-Kernighan. Esta abordagem pode ser entendida como um k-opt em que o algoritmo decide o valor de k para cada iteração com base em determinadas condições verificadas na solução (ver Lin e Kerninghan (1973) para uma descrição mais detalhada).

A primeira ILS para este problema foi proposta por Baum (1986b). Esta abordagem continha vários métodos de pesquisa local que escolhiam a primeira solução com menor custo encontrada na vizinhança (estratégia de *primeira-melhoria*). A perturbação consistia em aplicar uma iteração de 2-opt de um forma aleatória.

Martin et al. (1991b) apresentaram o primeiro ILS a obter um grande desempenho neste problema. Esta abordagem utilizava um método de pesquisa local baseado em 3-opt com estratégia de primeira-melhoria. Mais tarde, os mesmos autores observaram que o algoritmo de Lin-Kernighan era preferível ao 3-opt em termos de desempenho do ILS. A perturbação escolhida em todas abordagens foi a *double-bridge*, que consiste em efectuar uma só iteração de 4-opt, como é ilustrado na Figura 12.2. De acordo com os autores, esta perturbação só deve ser efectuada se a soma dos pesos das quatro arestas a remover é inferior a um valor proporcional ao peso médio das arestas da solução actual. Não só é impossível reproduzir esta perturbação através de pequeno número de iterações de 2-opt, 3-opt ou Lin-Kerningham, como o custo da solução perturbada não difere muito do custo da solução actual. Por esta razão, esta perturbação tem aparecido em vários estudos de ILS para este problema e tem sido extremamente eficaz. Finalmente, os autores utilizaram um critério de aceitação baseado em arrefecimento simulado. É necessário realçar que o bom desempenho do ILS final é também resultante das técnicas inovadoras e eficientes que foram propostas para a avaliação parcial da vizinhança sem grande prejuízo da qualidade final.

Uma variante da ILS descrita acima foi proposta por Johnson e McGeoch (1997), também denominada *Iterated Lin-Kernighan*. Nesta abordagem, a perturbação consistia de um movimento *double-bridge* sem as condições impostas nos pesos. O critério de aceitação seleccionava a solução com menor custo e a solução inicial era obtida por uma heurística gulosa aleatorizada. Os resultados obtidos por estes autores tornaram-se estado-da-arte para problemas do caixeiro viajante com milhares de cidades.

Outra abordagem semelhante foi proposta por Applegate et al. (2003). Esta ILS, denominada *Chained Lin-Kernighan*, utilizava também o método de pesquisa local de Lin-Kernighan, tal como os seus sucessores, mas considerava um número restrito de arestas. A perturbação é baseada em *double-bridge* mas só é aplicada a uma pequena parte da solução actual. As variantes seguintes reportadas por estes autores deram origem a um dos algoritmos mais rápidos para o problema do caixeiro viajante.[5] A implementação final consegue retornar soluções a 1% do óptimo para problemas com 25 milhões de cidades em menos de um dia de processamento num só computador pessoal.

[5] As implementações estão disponíveis em `http://www.tsp.gatech.edu/concorde.html`

A última versão mais conhecida do ILS para este problema foi proposta por Helsgaun (2000) e é conhecida por *Iterated Helsgaun Algorithm*. Esta ILS utiliza igualmente o método de pesquisa local de Lin-Kernighan. Contudo, diferente dos seus sucessores, utiliza uma perturbação que constrói uma solução com base na heurística de vizinho mais próximo, aproveitando algumas arestas da solução actual. O critério de aceitação também só aceita a solução com menor custo. Esta abordagem é actualmente umas das mais competitivas e encontra o óptimo para problemas com dezenas de milhares de cidades em poucos minutos.

4. Pesquisa de Vizinhança Variável

A metaheurística VNS utiliza um método de pesquisa local que alterna entre uma sequência de vizinhanças $N^1, N^2, \ldots, N^m$ (Mladenović e Hansen, 1997). De acordo com estes autores, esta abordagem explora três factos: i) um óptimo local para uma determinada vizinhança não é necessariamente um óptimo local para uma vizinhança diferente; por outras palavras, VNS está constantemente a modificar as bacias de atracção; ii) o óptimo global é também um óptimo local para qualquer uma das m noções de vizinhança; iii) em muitos problemas discretos verifica-se que óptimos locais para uma ou várias noções de vizinhança estão relativamente "perto" entre si. De realçar que este último facto é, na realidade, uma observação empírica mas que tem sido constatada por vários autores. Por exemplo, este parece ser o caso para o problema do caixeiro viajante para determinadas noções de vizinhança (Boese et al., 1994).

Na primeira versão de VNS, denominada *Descida em Vizinhança Variável* (VND), a mudança de vizinhança é efectuada a cada iteração do método de pesquisa local subordinado. Esta mudança segue uma determinada sequência definida *a priori*. A metaheurística termina quando não é possível encontrar uma solução vizinha com custo inferior para todas as vizinhanças. Numa versão de VNS mais avançada, denominada *VNS Reduzida*, a mudança para a vizinhança seguinte é efectuada se não existir uma solução vizinha com custo inferior. Normalmente, a solução vizinha é escolhida de um forma aleatória. A partir do momento que uma solução com menor custo é encontrada, o método de pesquisa local volta a utilizar a vizinhança que ocupa a primeira posição da sequência de vizinhanças. Esta abordagem tem grandes vantagens relativamente à anterior em termos de tempo computacional, pois evita a exploração total da vizinhança e permite que a primeira vizinhança, normalmente de menor complexidade, seja utilizada mais vezes. De notar que esta variante termina de acordo com algum critério definido pelo utilizador, iterações ou tempo computacional, dado que não é possível identificar um óptimo local quando a solução vizinha é escolhida aleatoriamente.

A abordagem mais conhecida actualmente é a *VNS Básica* e é a que se aproxima mais da ILS. Inicialmente, uma solução é gerada aleatoriamente ou de uma forma heurística. Esta é a solução inicial para um método de pesquisa local que utiliza a vizinhança N^1 e que retorna um óptimo local. Seguidamente, este procedimento é repetido por várias iterações em que a solução inicial é escolhida aleatoriamente na vizinhança do óptimo local da iteração anterior de acordo com uma vizinhança N^k previamente seleccionada. O critério de aceitação de VNS selecciona, entre o novo óptimo local e o anterior, aquele que tiver menor custo; se for o novo óptimo local, então a vizinhança seleccionada para a próxima próxima iteração será a que ocupa a primeira posição, caso contrário, será a vizinhança seguinte na sequência. É de realçar que o processo de gerar solução inicial em cada iteração da VNS Básica não é nada mais do que uma perturbação típica de ILS. A maior diferença entre as duas abordagens é que VNS Básica explora as noções de vizinhança de um modo mais explícito, sendo por isso um caso particular do ILS.

Existem muitas variantes de VNS na literatura. Por exemplo, é possível considerar diversas ordens na sequência de vizinhanças. Também é possível encontrar critérios de aceitação baseados em arrefecimento simulado ou com base numa determinada medida de distância entre soluções. Outra versão mais complexa é a *VNS de Decomposição* cuja ideia principal é aplicar o método de pesquisa local só em determinados componentes da solução.

5. Conclusões

Com este capítulo pretendeu-se introduzir, de um modo breve, os princípios básicos de ILS e VNS. Claramente, ainda há muito mais para ler e investigar. O leitor mais interessado poderá explorar os artigos que serviram de base para este texto tais como (Lourenço et al., 2002; Mladenović e Hansen, 1997) ou o livro de Hoos e Stützle (2004) e recolher informação mais actualizada em actas de conferências da *Metaheuristics International Conference*, *Stochastic Local Search Workshop* e *Learning and Intelligent Optimization Workshop*.

ILS ocupa um lugar invejável relativamente às outras metaheurísticas, não só pela sua simplicidade mas também no que diz respeito ao seu desempenho para o problema do caixeiro viajante. Contudo, é possível também encontrar abordagens de ILS com desempenho bastante competitivo para problemas de escalonamento, problemas de afectação quadrática, problema de coloração de grafos, problema de MAX-SAT e muitos mais. Existem ainda muitos problemas a serem abordados por esta metaheurística, entre os quais, problemas de optimização que lidam com múltiplos objectivos e em que os dados de entrada são estocásticos e/ou são fornecidos dinamicamente.

Dado o bom desempenho do ILS numa grande classe de problemas, é necessário entender a razão deste desempenho além da explicação intuitiva fornecida por este texto. Uma possibilidade é recolher informação pertinente sobre a estrutura do espaço de pesquisa, por exemplo, através de determinadas estatísticas do grafo subjacente ao tuplo *(problema, representação, vizinhança)*.

Finalmente, as interações entre os vários componentes de ILS e VNS necessitam de ser melhor entendidas. Existe algum *know-how* sobre como optimizar os vários componentes, mas não existe nenhuma metodologia que o faça de um modo sistemático para um dado problema. O recurso a técnicas estatísticas, tal com desenho experimental, permitem quantificar os efeitos e interações dos diversos componentes no desempenho global de ILS e VNS.