1 Clark and Wright

O algoritmo poupaça de Clarke e Wright é uma das mais conhecidas heuristicas para VRP. Foi desenvolvido em referenciar e aplica-se a problemas para os quais o numero de veículo não é fixo, e trabalha igualmente bem para problemas com digrafos e grafos. Quando duas rotas $(0, \ldots, i, 0)$ e $(0, j, \ldots, 0)$ podem possivelmente ser mescladas em uma rota simples $(0, \ldots, i, j, \ldots, 0)$, uma distância econômica $s_{ij} = c_{i0} + c_{0j} - c_{ij}$ é gerada. O algoritmo trabalha como segue:

Step 1. Savings computation

- Calcule a poupaça $s_{ij} = c_{i0} + c_{0j} - c_{ij}$ para $i, j = 1, \ldots, n$ e $i \neq j$.
- Crie n rotas de veículos $(0, i, 0)$ para $i = 1, \ldots, n$.
- Ordene as poupanças de modo não crescente

Step 2. Best feasible merge (Parallel version)

Inicie do topo da lista de poupanças, executando o seguinte:

- Dada uma poupança s_{ij}, determine se há duas rotas que podem possivelmente ser mescladas:
  - Uma iniciando com $(0, j)$
  - Uma terminando com $(0, i)$
- Combine essas duas rotas deletando $(0, j)$ e $(0, i)$ e introduzindo (i, j).

Step 2. Extensão de Rotas (Versão Sequêncial)

- Considere uma volta de cada rota $(0, i, \ldots, j, 0)$.
- Determine a primeira poupança s_{ki} ou s_{jl} que pode possivelmente ser usada para mesclar a rota atual com outra rota terminando com $(k, 0)$ ou iniciando com $(0, l)$.
- Implemente a mescla e repetição dessa operação para a rota atual.
- Se não há mesclas possiveis, considere a próxima rota e reaplique a mesma operação.
- Pare quando a mesclagem de rotas não for possivel.

2 Emparelhamento baseado no algoritmo de poupança

Ista é uma modificação interessante para o algoritmo de poupança padrão (descrições similares são feitas por referênciar e referênciar onde em cada interação a poupança s_{ij} obtida por mesclar rotas p e q é calculado como $s_{ij} = t(S_i) + t(S_j) - t(S_i \cap S_j)$, onde S_k é o conjunto de vértices da rota k, e $t(S_k)$ é o comprimento de uma solução ótima para o TSP em S_k.

Um problema de emparelhamento sobre o conjunto S_k é resolvido usando os s_{ij} valores com custo de emparelhamento, e as rotas correspondem à emparelhamento ótimo estão mescladas a permanencia da fazibilidade.

3 Algoritmo de melhoria para Mult-rota

Algoritmos de melhoria esforçam-se para atualizar alguma possivel solução executando um sequência de mudançãs dos vertices e arestas dentro ou entre rotas de veículos. Heuristicas de melhoramento de multi-rotas para VRP operão em cada rota de veículo tomando muitas rotas em um tempo(?). Podemos encontrar descrições de mudanças de arestas para VRP nestas três referências:

- Referenciar
- Referenciar
- Referenciar

Thompson e Psaraftis (1993) propõem um método baseado nos conceitos de k-transferência ciclicas que envolvem transferência de k demandas da rota I^j para a rota $I^{\delta(j)}$ para cada j e inteiro fixo k. O conjunto de rotas $\{I^r\}$, com $r = 1, \ldots, m$, constitui um solução possivel e δ é uma permutação cíclica de um subconjunto de $\{1, \ldots, m\}$. Em particular, quando δ tem cardinalidade fixa C, obtemos um C-ciclo k-transferência. Permitindo k demanda modelo em cada rota, transferência de demanda pode ser realizada através de permutações mais que permutações ciclicas, Devido a complexidade das pesquisas de vizinhanças de transferências cíclicas, é realizada heuristicamente. O operador de 3-ciclo 2-transferências é ilustrado na figura abaixo.

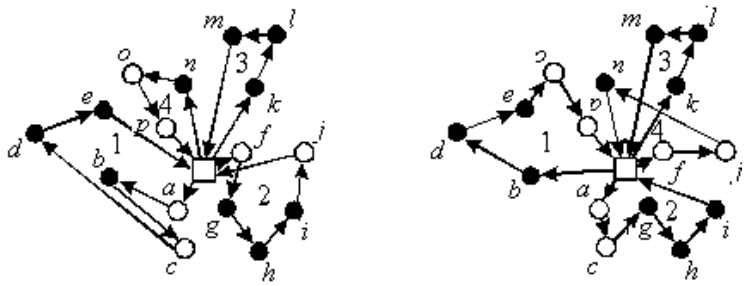
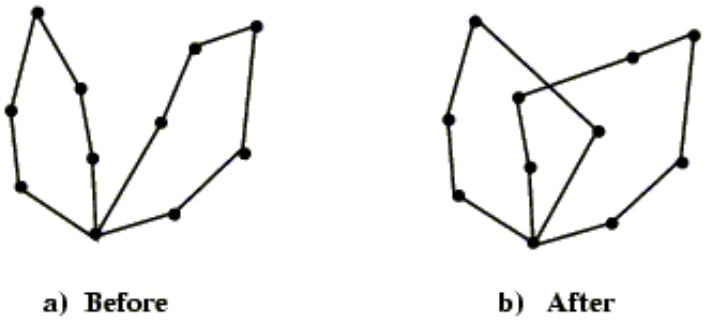


Figura 1: O operador de transferência cíclica. A ideia base é transferir simultaneamente os clientes denotados pelo ciclo branco de maneira ciclica entre as rotas. Mais precisamente aqui clientes a e c na rota 1, f e j na rota 2 e o e p na rota 4 são transferidos simultaneamente para as rotas 2, 4, e 1 respectivamente e a rota 3 permanece intocada.

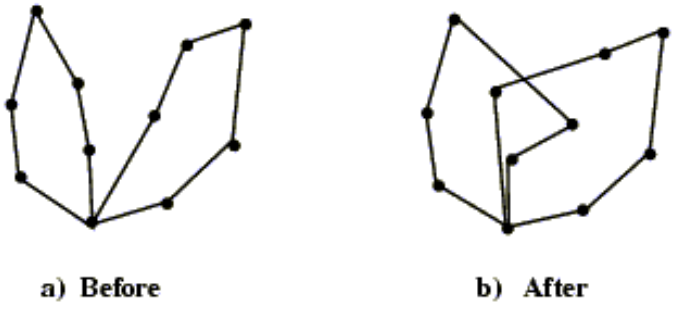
3.1 VAN BREEDAM'S ANALYSIS

Agora sumarizaremos a analise de Van Breedam's. Há quatro operações a considerar, as quais são:

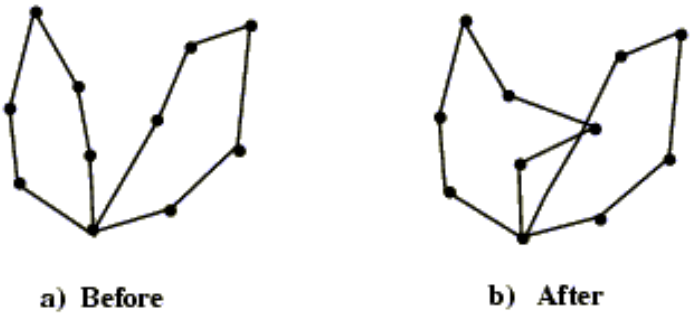
1. String Cross (SC): Duas cadeias de vertices são mudadas pelo cruzamento de duas arestas de diferêntes rotas.
2. String Exchange (SE): Duas cadeias de no mínimo k vértices são mudadas entre duas rotas.



a) Before b) After



a) Before b) After



a) Before b) After

3. String Realocação (SR): Uma cadeia de no mínimo k vertices é deslocada de uma rota para outra, tipicamente com $k = 1$ ou 2.

4. String Mix (SM): O melhor movimento entre SE e SR é selecionado.

   Para avaliar estes movimentos, Van Breedam considerou duas estratégias de melhoria local:

   (a) First Improvement (FI): Consiste de implementar o primeiro movimento que melhora a função objetivo.

   (b) Best Improvement (BI): Avalia todas os possiveis movimentos e implementa o melhor destes.

   Van Breedam então define um conjunto de paramentros que pode influenciar o comportamento da produção da melhoria local:

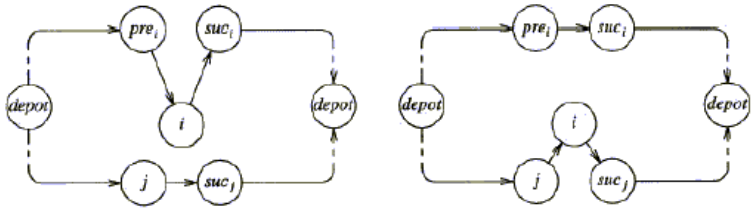
   - A solução inicial (pobre, bom)
   - O comprimenro da string (k) para movimentos do tipo SE, SR, SM (k=1 ou 2)
   - A estratégia de selação (FI, BI)

- O processo de avaliação para uma string de comprimento $k > 1$ (avaliar todas as possiveis strings de comprimento entre um de rotas, crescentod k quando uma avaliação de ciclo inteiro tem sido completado sem identificar um mevimento de melhoria).

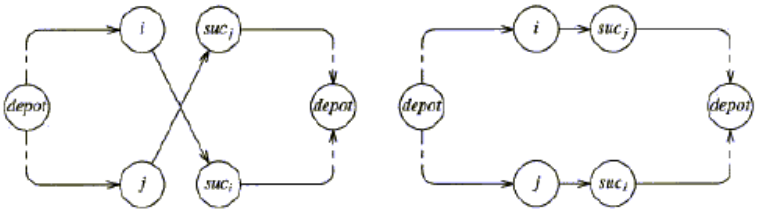
3.2 KINDERWATER AND SAVELSBERGH

Na heuristica de Kinderwater e Savelsbergh rotas não são isoladas, então caminhos e clientes são mudados entre diferêntes rotas. A operação que faz essas mudanças são:

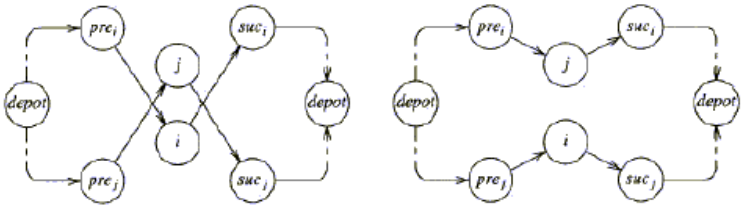
1. Customer Relocation: Um cliente localizado em uma rota é mudado para outra



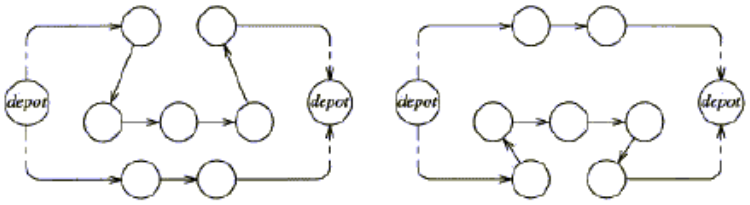
2. Crossover: Duas rotas são misturadas em um ponto



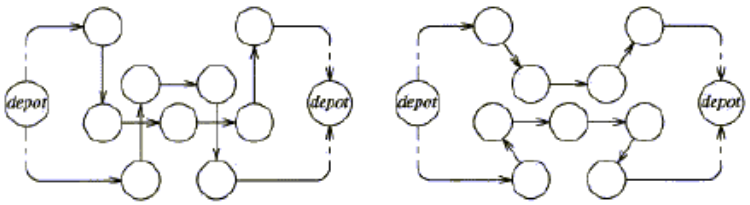
3. Customer Exchamge: Dois clientes de ruas rotas diferêntes são mudados entre duas rotas



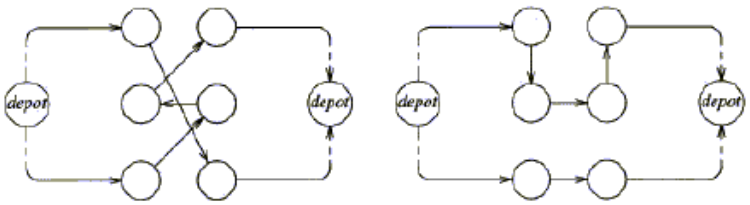
Nas figuras seguintes podemos ver um exemplor um pouco mais complexo:



(a) Relocation of a path.



(b) Exchange of two paths.



(c) A crossover plus 2-exchange.