

ESTUDO DA INCORPORAÇÃO DE *CROSS-DOCKING* AO PROBLEMA DE ROTEAMENTO DE VEÍCULOS

Vinicius W. C. Moraes

Universidade Federal de Minas Gerais (UFMG)
Av. Antônio Carlos, 6627 – Pampulha – Belo Horizonte – MG – CEP 31270-901
vwcmorais@dcc.ufmg.br

Geraldo Robson Mateus

Universidade Federal de Minas Gerais (UFMG)
Av. Antônio Carlos, 6627 – Pampulha – Belo Horizonte – MG – CEP 31270-901
mateus@dcc.ufmg.br

Thiago F. Noronha

Universidade Federal de Minas Gerais (UFMG)
Av. Antônio Carlos, 6627 – Pampulha – Belo Horizonte – MG – CEP 31270-901
tfn@dcc.ufmg.br

RESUMO

O Problema de Roteamento de Veículos com *Cross-Docking* integra ao Problema de Roteamento de Veículos (PRV) uma etapa de consolidação. Dado um conjunto de consumidores e fornecedores, com suas respectivas demandas de coleta ou entrega e um único centro de consolidação (CD), o problema visa minimizar o somatório dos custos das rotas de cada veículo, que coletam os produtos nos fornecedores e os entregam aos consumidores passando pelo CD. No CD os produtos podem ou não ser realocados entre os veículos, processo este conhecido como consolidação. O objetivo deste trabalho é verificar se a adoção do processo de consolidação no CD possibilita obter ganhos ao definir o roteamento dos veículos. Assim apresentamos algoritmos baseados na metaheurística ILS que tratam o problema nas versões com e sem consolidação. Testes foram realizados para instâncias com 30, 50, 100, 150 e 200 requisições, obtendo resultado satisfatórios que comprovaram a eficiência da incorporação da etapa de *cross-docking* à atividade de roteamento.

PALAVRAS CHAVE. Logística & Transportes, Metaheurísticas, Otimização Combinatória. Área principal. Logística & Transportes.

ABSTRACT

This paper addresses the Vehicle Routing Problem with Cross-Docking (VRPCD). Given a set of suppliers and consumers, with their respective demands of pickup and delivery and a single Cross-Dock (CD), the problem aims to minimize the transportation cost for each vehicle, which pickup the products from suppliers and deliver to consumers throughout CD, where they can be consolidated. The goal of this work is to verify if the addition of the consolidation process at the cross-dock improves the routing activity. Therefore we propose an ILS-based algorithm to solve the problem in both versions, i.e., with or without consolidation at the cross-dock. Tests were run on instances with 30, 50, 100, 150 and 200 requests, and good results were achieved, showing the efficiency of adding the Cross-Docking to the routing process.

KEYWORDS. Logistics & Transport, Metaheuristics, Combinatorial Optimization.
Main area. Logistics & Transport.

1. Introdução

O sistema de produção que envolve a coleta de matéria prima, manufatura, armazenamento e entrega de produtos aos consumidores finais é conhecido como Cadeia de Suprimentos (do inglês *Supply Chain*). Uma das mais importantes atividades dessa cadeia é o transporte de produtos de um ponto de origem a um ponto de destino. Neste contexto, surgem diversos problemas explorados em Pesquisa Operacional, sendo um deles o Problema de Roteamento de Veículos (VRP, do inglês *Vehicle Routing Problem*).

Dado um conjunto de consumidores dispersos geograficamente, com suas demandas não negativas e uma frota de veículos homogêneos com capacidade limitada, o objetivo do VRP é definir rotas de custo mínimo partindo de um depósito central e visitando cada consumidor uma única vez. Proposto por Dantzig e Ramser (1959) como uma generalização do Problema do Caixeiro Viajante (TSP, do inglês *Traveling Salesman Problem*), o VRP dá origem a diversas variantes com um grande número de trabalhos publicados na literatura.

Uma importante variante do VRP é o Problema de Roteamento de Veículos com Coleta e Entrega (VRPPD, do inglês *Vehicle Routing Problem with Pickup and Delivery*), onde as cargas devem ser transportadas de pontos de coleta a pontos de entrega [Parragh et al. (2008)]. Outra importante extensão do VRP é o Problema de Roteamento de veículos com Janela de Tempo (VRPTW, do inglês *Vehicle Routing Problem with Time Windows*). No VRPTW cada consumidor deve ser visitado apenas uma vez dentro de um intervalo de tempo específico ou janela de tempo. Existem duas versões do VRPTW que são estudadas na literatura, cujos objetivos são: (i) definir rotas de custo mínimo ou duração mínima, respeitando as janelas de tempo e (ii) definir um número mínimo de veículos para atender todas as demandas [Solomon (1987); Bräysy e Gendreau (2005a,b)].

Na cadeia de suprimentos um importante conceito é o de *Cross-docking*. Este é usado para gerir o fluxo de um sistema de transporte e reduzir o custo associado ao depósito de materiais na rede logística [Jayaraman e Ross (2003)]. Nesta abordagem um conjunto de produtos vindos de diferentes origens chega a um centro de consolidação (CD, do inglês *Cross-dock*), onde as cargas são realocadas entre os veículos e enviadas aos consumidores finais de forma a minimizar os custos de transporte. O processo de carga, descarga e agendamento de requisições no CD é conhecido como *consolidação*. A atividade no CD é considerada constante, uma vez que os produtos não permanecem estocados por um período superior a um dia de trabalho [Yu e Egbelu (2008)].

O Problema de Roteamento de Veículos com *Cross-Docking* (VRPCD, do inglês *Vehicle Routing Problem with Cross-Docking*) acrescenta aos Problemas de Roteamento Veículos uma etapa de *Cross-Docking*. O VRPCD pode ser definido como segue. Dado um grafo direcionado $G(N,E)$, em que N é o conjunto de nós formado pela união dos conjuntos de fornecedores $P = \{p_1, p_2, \dots, p_n\}$ e consumidores $D = \{d_1, d_2, \dots, d_n\}$, além do centro de consolidação CD , $N = P \cup D \cup \{CD\}$, e E é o conjunto de arcos. Seja R um conjunto com n requisições, $R = \{r_1, r_2, \dots, r_n\}$, dado que cada $r_i \in R$ é associado a um par de nós $\{p_i, d_i\}$, tal que, $p_i \in P$ e $d_i \in D$; Dado ainda o custo c_{ij} associado a cada arco $(i,j) \in E$ e $[a_i, b_i]$ a janela de tempo do nó $i \in N$. O objetivo do VRPCD é definir rotas de custo mínimo para uma frota de veículos homogêneos de capacidade Q , satisfazendo as demandas de cada nó ($\forall i \in P \cup D$), coletando as requisições nos fornecedores e enviando aos consumidores passando pelo CD, onde a cada operação de carga ou descarga é associado um tempo proporcional a cada unidade de produto consolidada.

No VRPCD dois contextos podem ser visualizados, conforme ilustrado nas Figuras 1 e 2. No primeiro caso (Fig. 1) é permitido a um veículo realizar coleta e entrega passando pelo CD, porém sem realizar o processo de consolidação. Já o segundo caso (Fig. 2), após efetuarem a coleta, os veículos retornam ao CD, onde passam pelo processo de consolidação, antes de iniciarem o processo de entrega [Musa et al. (2010)].

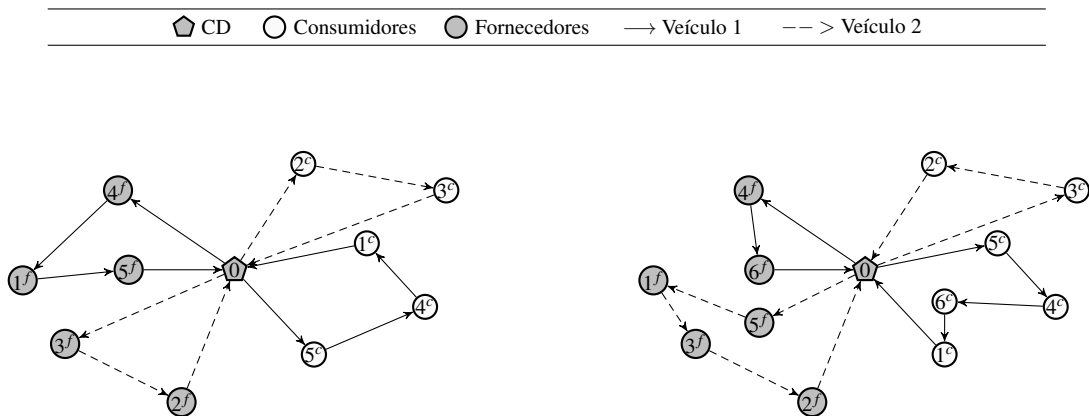


Figura 1: Primeiro caso do VRPCD. O veículo 1 visita os fornecedores 4, 1 e 5, em seguida entrega os produtos aos consumidores 5, 4, e 1 passando pelo CD sem realizar o processo de consolidação. O veículo 2 visita os fornecedores 3 e 2, em seguida entrega diretamente os produtos aos consumidores 2 e 3 sem realizar consolidação no CD.

Figura 2: Segundo caso do VRPCD. O veículo 1 visita os fornecedores 4 e 6, retorna ao CD, onde realiza a consolidação e em seguida entrega os produtos aos consumidores 5, 4, 6 e 1. O veículo 2 coleta os produtos nos fornecedores 5, 1, 3 e 2, retorna ao CD, onde realiza a consolidação e em seguida entrega os produtos aos consumidores 3 e 2.

Este trabalho apresenta dois algoritmos baseados na metaheurística ILS [Lourenco et al. (2003)] que tratam o problema VRPCD nas versões com e sem consolidação. Na Seção 2 são descritos os trabalhos relacionados. Os algoritmos propostos são apresentados na Seção 3, onde descrevemos os algoritmos de construção da solução inicial, procedimentos de busca local e o método de perturbação utilizado. Resultados experimentais são apresentados na Seção 4. Por fim, as considerações finais são expostas na Seção 5.

2. Trabalhos relacionados

O primeiro trabalho proposto para o VRPCD foi o de Lee et al. (2006), que o resolveram por meio de um algoritmo baseado em Busca Tabu. Nesta versão é considerado um único centro de consolidação, de onde os veículos partem. Assim como no VRPTW cada ponto deve ser atendido dentro de uma janela de tempo e toda a atividade de roteamento e consolidação deve ser realizada respeitando um horizonte de planejamento. No VRPCD apresentado, os veículos de coleta partem do CD em direção aos fornecedores, coletam as demandas e retornam simultaneamente ao ponto inicial para realizar o processo de consolidação, posteriormente, os veículos visitam os consumidores, satisfazendo suas demandas.

Liao et al. (2010) abordam o problema na mesma versão apresentada em Lee et al. (2006) e também propuseram um algoritmo baseado em Busca Tabu. As principais diferenças entre esses dois trabalhos são suas heurísticas para geração da solução inicial e os mecanismos de busca local. Em Liao et al. (2010) é utilizado o operador de busca local *Reallocate*, em que, consumidores são removidos de um veículo e atribuídos a outro, enquanto o algoritmo de Lee

et al. (2006) é baseado no operador *Exchange*. Outra diferença é que apenas o algoritmo de Liao et al. (2010) admite decrementar o número de veículos utilizados durante a execução do algoritmo. Os experimentos foram realizados sob o mesmo conjunto de instâncias, para as quais o algoritmo de Liao et al. (2010) obteve melhores resultados.

Em Wen et al. (2009) é apresentada uma versão do VRPCD diferente daquela proposta por Lee et al. (2006). Nesta versão os veículos passam pelo o processo de consolidação tão logo terminem de realizar a coleta, sem a necessidade de todos estarem no CD simultaneamente. A etapa de consolidação demanda um tempo para carga e descarga de produtos no CD. Tempo este que é refletido no cálculo da janela de tempo na atividade de entrega. O objetivo do problema é definir um conjunto de rotas de coleta e entrega de custo mínimo passando obrigatoriamente pelo CD, onde realizam consolidação. Foram propostos um conjunto de instâncias, uma formulação matemática para o problema e uma eficiente heurística baseada em Busca Tabu.

Santos et al. (2010, 2011a,b) apresentaram uma variação do VRPCD, onde é considerado o custo fixo em se manipular as mercadorias no centro de consolidação, porém o roteamento não é limitado pelas restrições de janela de tempo. As principais contribuições de Santos et al. (2010) foram as propostas de modelos matemáticos e algoritmos exatos baseados nestas formulações, que foram aplicados a instâncias adaptadas daquelas propostas em Solomon (1987) e Wen et al. (2009). Em Santos et al. (2011a) e Santos et al. (2011b), respectivamente, são apresentadas uma formulação por geração de colunas e um algoritmo *branch-and-price* para o problema, obtendo soluções ótimas para instâncias com até 30 requisições.

Musa et al. (2010) direcionam seus trabalhos ao problema de redes de *Cross-docking* onde cargas são transferidas de pontos origem para pontos de destino passando por facilidades de *cross-docking*. Neste problema busca-se minimizar o custo de transporte na rede, carregando os caminhões nos fornecedores e os levando aos consumidores diretamente ou indiretamente por meio de centros de consolidação. A principal diferença deste trabalho com relação ao VRPCD tratado neste e no trabalho de Wen et al. (2009) é que no contexto explorado por Musa et al. (2010) a solução para o problema é um conjunto de rotas abertas que cobrem pares de pontos origem e destino, em que, os veículos não iniciam e terminam suas rotas nos pontos centrais.

3. Estratégia de Solução

Dado que o VRPCD é NP-Difícil [Lee et al. (2006)], métodos heurísticos têm sido aplicados para encontrar soluções em um curto período de tempo. A metaheurística *Iterated Local Search* (ILS) é um procedimento que parte de uma solução viável e aplica sucessivamente busca local e perturbação, limitando o espaço de busca a um conjunto de ótimos locais [Lourenco et al. (2003)]. Na Subseção 3.1 é descrita a heurística proposta para geração da solução inicial, posteriormente, apresentamos os algoritmos baseados em ILS na Subseção 3.2.

3.1. Solução inicial

O método para geração de solução inicial proposto é um algoritmo baseado na heurística construtiva *Nearest Insertion* (NI) para o TSP. Seja d_i um consumidor de uma requisição atribuída à uma rota de entrega e, d_j outro consumidor de uma requisição ainda não atribuída à nenhuma rota, é dado o custo c_{d_i, d_j} do arco (d_i, d_j) . Seja ainda, p_i um fornecedor de uma requisição atribuída à uma rota de coleta e p_j um fornecedor de uma requisição ainda não atribuída, é dado o custo c_{p_i, p_j} do arco (p_i, p_j) . Na heurística *NI-CD*, proposta neste trabalho, as

requisições são sequencialmente adicionadas à solução, ordenadas pelo valor Δ_c que é igual a $c_{d_i,d_j} + c_{p_i,p_j}$, sem considerar o processo de consolidação no CD.

A Figura 3 ilustra o procedimento construtivo proposto. Dadas as rotas que atendem as requisições 1, 2 e 3, é verificada a requisição ainda não atribuída a nenhum veículo com o menor valor Δ_c . Note que no exemplo a requisição escolhida e atribuída ao veículo é a 4.

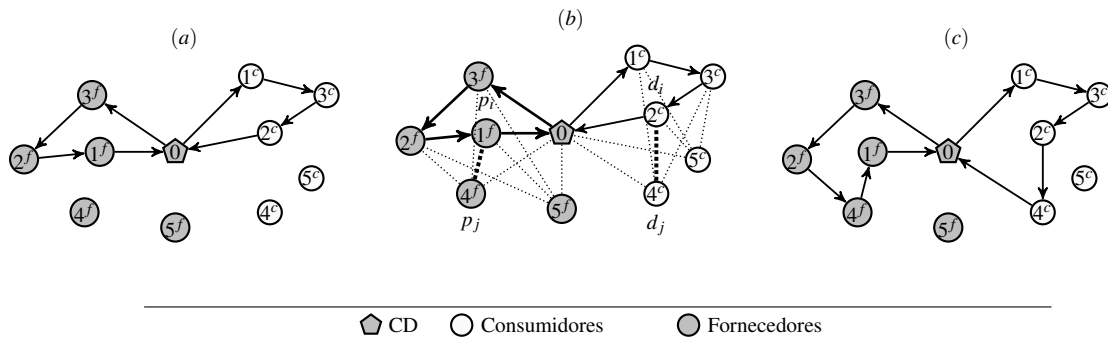


Figura 3: Ilustração da heurística NI-CD. Para cada requisição ainda não atribuída é calculado o valor Δ_c em relação a cada nó (consumidor e fornecedor) das requisições cujos nós já foram roteados. A Figura (a) mostra o estado da solução em que as requisições 1, 2 e 3 já foram atribuídas ao veículo. O próximo passo do algoritmo, demonstrado na Figura (b), é buscar a próxima requisição ainda não atribuída para ser alocada ao veículo. Na Figura (c), a requisição 4 é atribuída ao veículo, por ser a de menor Δ_c .

Os passos da heurística NI-CD são descritos na Figura 4. Na linha 1 é criada a lista de requisições ainda não atribuídas a nenhum veículo; nas Linhas 2–7 uma requisição é escolhida de forma aleatória e atribuída a um veículo, definindo suas rotas de coleta e entrega, respectivamente. No *loop* das linhas 8–18 a requisição ainda não atribuída de menor Δ_c é adicionada a um veículo.

```

início NI-CD(número de requisições N, grafo G, lista de demandas L): S
1.  $LNA \leftarrow$  Cria lista com as N requisições não atribuídas;
2.  $r \leftarrow$  Seleciona uma requisição aleatória de LNA;
3. Remove a requisição  $r$  da lista LNA;
4.  $RC \leftarrow$  Cria nova rota de coleta;
5.  $RE \leftarrow$  Cria nova rota de entrega;
6. Insere o nó fornecedor de  $r$  na rota de coleta RC;
7. Insere o nó consumidor de  $r$  na rota de entrega RE;
8. enquanto existir requisições na lista LNA faça
9.    $r \leftarrow$  Retorna a requisição em LNA de menor  $\Delta_c$ ;
10.  Remove a requisição  $r$  da lista LNA;
11.  Insere o nó fornecedor de  $r$  na rota de coleta RC;
12.  Insere o nó consumidor de  $r$  na rota de entrega RE;
13.  se a capacidade do veículo extrapolar então
14.    Insere as rotas RC e RE na solução S;
15.     $RC \leftarrow$  Cria nova rota de coleta;
16.     $RE \leftarrow$  Cria nova rota de entrega;
17.  fim-se
18. fim-enquanto;
19. retorna S;
fim
  
```

Figura 4: Pseudo-código da Heurística NI-CD.

3.2. ILS-VRPCD

Neste trabalho foram implementadas duas versões do algoritmo ILS para resolver as versões do VRPCD com e sem consolidação. O ILS-VRPCD-1 e o ILS-VRPCD-2 são descritos nas Subseções 3.2.1 e 3.2.2, respectivamente. As diferenças entre estas versões são os operadores de perturbação e busca local aplicados.

A Figura 5 ilustra o pseudo-código dos algoritmos ILS para o VRPCD. A solução inicial S é gerada na Linha 1 pela heurística construtiva NI-CD. Na Linha 2, à solução é aplicado o operador de busca local que retorna a melhor solução conhecida a S^* , a solução S é atualizada na Linha 3. No laço das Linhas 4–9, aplica-se sucessivamente perturbação e busca local sob a solução S , a fim de gerar um novo ótimo local S' . A melhor solução local S é atualizada pela solução S'' de acordo com o critério de aceitação definido na Linha 7. S'' é aceita se, e somente se, seu custo não for maior que 5% do custo de S .

```

begin ILS-CD(número de requisições N, grafo G, lista de demandas L):  $S^*$ 
1.  $S \leftarrow$  NI-CD(N, G, L);
2.  $S^* \leftarrow$  Busca Local( $S$ );
3.  $S \leftarrow S^*$ ;
4. enquanto critério de parada não satisfeito faça
5.    $S' \leftarrow$  Perturbação( $S$ );
6.    $S'' \leftarrow$  Busca Local( $S'$ );
7.   se CritérioDeAceitação( $S, S''$ ) então  $S \leftarrow S''$ ;
8.   se  $f(S'')$  é melhor que  $f(S^*)$  então  $S^* \leftarrow S''$ ;
9. fim-enquanto;
10. retorna  $S^*$ ;
fim

```

Figura 5: Pseudo-código do algoritmo ILS.

3.2.1. ILS-VRPCD-1

O ILS-VRPCD-1 resolve o primeiro caso do VRPCD. Dada uma solução inicial viável, os algoritmos de *busca local* geram iterativamente um conjunto de soluções vizinhas pela alteração de elementos que compõem uma solução corrente. Para o primeiro caso o VRPCD esta alteração se dá pela remoção de uma requisição de um veículo e a atribuição desta a outro. Foram implementados dois operadores de busca local, o *CD-Relocate* e o *CD-Exchange*.

O *CD-Relocate* percorre uma vizinhança de tamanho $O(n^3)$ por meio de trocas inter-veiculares. Neste operador uma requisição $r_i \in R$ é removida de um veículo v_i e atribuída a outro veículo v_j . Este movimento foi implementado sob a estratégia *melhor-aprimorante*.

O operador *CD-Exchange* itera sob uma vizinhança $O(n^4)$ trocando requisições entre pares de veículos. Neste operador uma requisição $r_i \in R$ é removida de um veículo v_i e atribuída a outro veículo v_j , enquanto uma requisição $r_j \in R$ é removida do veículo v_j e atribuída ao veículo v_i . O procedimento foi implementado na versão *primeiro-aprimorante*, ou seja, a primeira solução melhorante encontrada pelo algoritmo é retornada.

Uma *perturbação* gera um vizinho aleatório em uma vizinhança, que em geral é muito grande para ser explorada por um algoritmo de busca local. O *CD-Ejection Chain* é o procedimento de perturbação desenvolvido para o algoritmo ILS-VRPCD-1, baseado na perturbação *Ejection Chain* aplicada ao VRP por Rego e Roucairol (1996). Neste operador uma requisição é

escolhida de forma aleatória de um veículo r_1 e transferida para o veículo r_2 , depois uma requisição de r_2 é transferida para o veículo r_3 , e assim sucessivamente. O procedimento é executado até que seja transferida a requisição do último veículo para o veículo r_1 . O ciclo do algoritmo é executado λ vezes, onde λ é o parâmetro da perturbação.

3.2.2. ILS-VRPCD-2

O segundo caso do VRPCD é resolvido pelo algoritmo ILS-VRPCD-2. Neste algoritmo os operadores realizam trocas específicas nas rotas de coleta de um veículo independente de suas rotas de entrega, e vice-versa. A cada movimento é verificada a viabilidade do problema e atualizado o processo de trocas no centro de consolidação. Na fase de busca local do ILS-VRPCD-2 são utilizados dois operadores de busca local, o *CD Swap(1,1)* e o *CD Re-Insertion*.

O operador *CD-Swap(1,1)* efetua movimentos inter-rotas. Em rotas de coleta são efetuadas trocas entre um fornecedor p_i da rota r_i e um fornecedor p_j da rota r_j . Ao realizar trocas entre as rotas de coleta é necessário verificar a viabilidade das rotas de entrega, uma vez que o tempo de troca no CD pode inviabilizar a entrega pelas restrições de janela de tempo. Em rotas de entrega, realizam-se trocas entre um consumidor d_i da rota r_i e um consumidor d_j da rota r_j , porém só é necessário verificar a viabilidade entre as rotas de entrega.

O operador *CD-Re-insertion* realiza movimentos intra-rota. Nesta busca local, para cada rota de coleta e entrega, remove-se um nó de uma rota e o re-insere em outra posição da mesma rota. Neste operador, como não se realizam trocas entre rotas distintas, não é necessário verificar a viabilidade do processo de consolidação.

O operador de perturbação *CD-Ejection Chain* é estendido para algoritmo ILS-VRPCD-2. Neste mecanismo um nó escolhido de forma aleatória da rota de coleta r_1 é transferido para a rota de coleta r_2 , um nó da rota r_2 é transferido para a rota r_3 , e assim sucessivamente. O procedimento é repetido de forma similar para as rotas entrega.

4. Resultados Computacionais

Os algoritmos foram implementados em C++ e compilados com GCC v4.4.3. Os experimentos foram executados em um computador com processador Intel(R) Core(TM) 2 Duo CPU com 3.00 GHz com 4 GB de memória RAM, executando em Ubuntu 10.04 LTS 64 bits, com apenas um CPU core.

As instâncias de testes utilizadas foram propostas por Wen et al. (2009), com grupos de 30, 50, 100, 150 e 200 requisições. Experimentos foram realizados para os algoritmos ILS-VRPCD-1 e ILS-VRPCD-2. A condição de parada para os dois algoritmos foi o tempo, definido em 10 minutos. O critério de aceitação foi de 5% do custo da solução S'' e o parâmetro de perturbação λ foi fixado em 30% do número de requisições de cada instância.

Os resultados encontrados são apresentados na Tabela 1. As colunas 1 e 2 descrevem as instâncias utilizadas, onde n é o número de nós consumidores e fornecedores, a coluna 3 informa o limite inferior LB para cada instância, dado pela relaxação linear do 2-VRPTW, descrito em Wen et al. (2009). As colunas 4 e 5, apresentam o melhor resultado e a média para 25 execuções do algoritmo ILS-VRPCD-1, enquanto as colunas 7 e 8 mostram os resultados

para o algoritmo ILS-VRPCD-2, o *gap* percentual entre os melhores resultados dos algoritmos implementados e o LB são apresentados nas colunas 6 e 9, respectivamente.

Tabela 1: Resultados obtidos para as instâncias de Wen et al. (2009).

Problemas		Wen et al.	ILS-VRPCD-1			ILS-VRPCD-2		
Instância	n	LB	Melhor	Média	Gap	Melhor	Média	Gap
30a	60	3757.04	4174.17	4317.62	6.22	3929.84	4014.14	4.60
30b	60	4795.65	5402.74	5616.78	9.45	4936.36	5062.90	2.93
30c	60	4968.3	5661.36	5800.34	10.20	5137.52	5165.07	3.41
30d	60	3708.37	4387.43	4543.08	12.04	3915.97	4005.98	5.60
30e	60	4913.24	5703.01	5849.86	12.07	5089.00	5206.82	3.58
50a	100	6340.9	7319.08	7469.53	10.13	6645.82	6790.92	4.81
50b	100	7201.89	8547.36	8676.60	13.40	7537.17	7629.96	4.66
50c	100	7241.05	8124.96	8282.56	7.26	7575.00	7671.82	4.61
50d	100	6887.93	8025.25	8284.83	11.67	7186.45	7307.52	4.33
50e	100	7347.54	8644.74	8876.37	11.85	7728.61	7812.08	5.19
100a	200	12555.57	14937.42	15105.08	11.61	13383.28	13486.26	6.59
100b	200	14200.48	17003.25	17244.16	15.69	14697.28	15118.56	3.50
100c	200	13631.24	16163.29	16381.11	12.60	14355.18	14610.89	5.31
100d	200	13395.33	16233.62	16364.72	14.76	14146.04	14424.59	5.60
100e	200	13745.6	16476.97	16585.26	13.27	14546.79	14718.84	5.83
150a	300	19012.02	22988.75	23024.31	13.82	20197.25	20506.74	6.23
150b	300	20371.08	23916.32	24072.43	11.49	21450.78	21745.37	5.30
150c	300	19419.55	23555.30	23820.11	14.59	20555.79	20982.65	5.85
150d	300	20013.37	23946.23	24197.69	12.12	21357.56	21454.49	6.72
150e	300	19141.66	23143.78	23468.62	14.08	20286.66	20506.19	5.98
200a	400	26538.53	31598.83	31720.94	12.29	28140.23	28290.25	6.04
200b	400	26722.88	31828.25	32065.88	12.62	28261.99	28434.52	5.76
200c	400	25607.31	31308.64	31492.57	15.18	27181.92	27404.37	6.15
200d	400	26969.42	32061.81	32355.67	12.31	28546.85	28753.52	5.85
200e	400	25776.01	31279.77	31492.91	13.28	27613.42	27936.71	7.13
Média	-	-	-	-	12.16	-	-	5,26

Uma análise empírica foi realizada para mostrar a convergência do algoritmo ILS avaliando o processo de consolidação. A análise foi realizada sob a instância 100b e os resultados plotados no gráfico da Figura 6. Como pode ser observado, o método converge rapidamente até os 150 segundos, posteriormente, converge lentamente até alcançar o critério de parada.

Ao observar a tabela de resultados pode-se verificar que a adoção do processo de consolidação possibilitou um maior ganho em termos de minimização do custo de rotas, dado que a média do *gap* para o algoritmo ILS-VRPCD-2 foi de 5,26%, enquanto, a do algoritmo ILS-VRPCD-1 foi de 12,16%. Isso é evidenciado principalmente pelo fato do problema tratado penalizar as trocas durante o processo de consolidação apenas na janela de tempo e não no

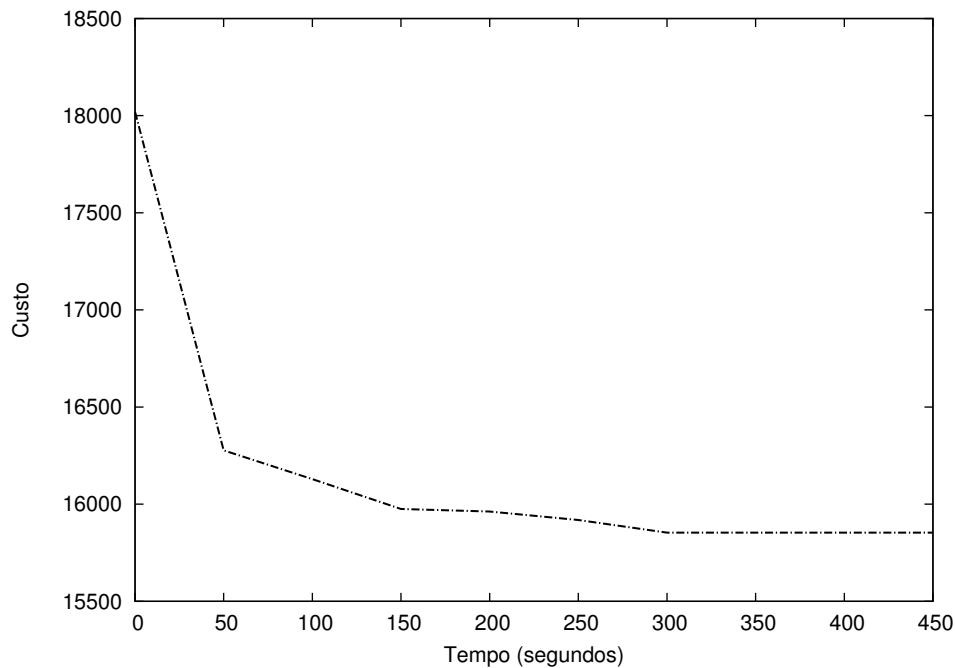


Figura 6: Teste de convergência do algoritmo ILS sob a instância 100b ao resolver o VRPCD considerando o processo de consolidação no CD. *Note* que o método converge rapidamente até os 150 segundos de execução, obtendo pouca melhora após esse período.

custo objetivo. Assim, ao se manter a viabilidade de tempo, em algumas instâncias poderia otimizar de forma independente tanto o processo de coleta quanto o de entrega.

5. Considerações Finais

Este trabalho aborda o Problema de Roteamento de Veículos com *Cross-Docking* (VRPCD, do inglês *Vehicle Routing Problem with Cross-Docking*). Foram tratados dois cenários para o problema, o primeiro em que os veículos coletam e entregam os produtos passando diretamente pelo centro de *Cross-dock* (CD) sem realizar qualquer tipo de consolidação e, o segundo em que os veículos passam por consolidação no CD tão logo realizem a coleta dos produtos nos fornecedores. Para tratar o problema foram apresentadas duas versões de um algoritmo baseado metaheurística ILS, quatro operadores de busca local e uma de perturbação.

O objetivo principal deste trabalho foi verificar se a incorporação da etapa de *Cross-Docking* ao processo de roteamento é benéfica. Respondendo a hipótese levantada durante o desenvolvimento deste trabalho, foi verificado que a adoção do processo de consolidação pode trazer ganhos consideráveis a todo o processo de transporte. Este ganho ocorre principalmente para os casos onde o custo de consolidação no CD é baixo comparado aos custos de roteamento.

Nossos estudos apontam a integração da etapa de consolidação aos problemas de VRP como uma boa alternativa para cenários em que ocorrem processos de coleta e entrega em pontos distintos. Outro fato a se notar é que no contexto do problema de coleta e entrega tratado neste trabalho, a utilização do centro de consolidação possibilita realizar o transporte de produtos com uma frota menor de veículos, uma vez que a otimização do processo de consolidação permitirá definir melhores rotas de entrega.

Os resultados comprovaram a importância dos nossos experimentos por mostrar a diferença em se realizar ou não processo de consolidação, com o intuito de melhor aproveitar a frota veicular e reduzir os custos do transporte das requisições pela rede de consumo. Pretende-se em trabalhos futuros, implementar novas estruturas de vizinhança, otimizar as estruturas de dados usadas, além de propor um novo conjunto de instâncias para o problema.

Agradecimentos

Este trabalho foi financiado pela Coordenação de Aperfeiçoamento de Pessoal de Nível Superior, CAPES.

Referências

- Bräysy, O. e Gendreau, M.**, Vehicle routing problem with time windows, part i: Route construction and local search algorithms, *Transportation Science*, 39:104–118, February 2005a, ISSN 1526-5447.
- Bräysy, O. e Gendreau, M.**, Vehicle routing problem with time windows, part ii: Metaheuristics, *Transportation Science*, 39:119–139, February 2005b, ISSN 1526-5447.
- Dantzig, G. B. e Ramser, J. H.**, The truck dispatching problem, *Management Science*, (6): 80–91, 1959.
- Jayaraman, V. e Ross, A.**, A simulated annealing methodology to distribution network design and management, *European Journal of Operational Research*, 144(3):629–645, 2003.
- Lee, Y. H., Jung, J. W., e Lee, K. M.**, Vehicle routing scheduling for cross-docking in the supply chain, *Computers and Industrial Engineering*, 51:247–256, October 2006, ISSN 0360-8352.
- Liao, C.-J., Lin, Y., e Shih, S. C.**, Vehicle routing with cross-docking in the supply chain, *Expert Systems with Applications*, 37(10):6868 – 6873, 2010, ISSN 0957-4174.
- Lourenco, H., Martin, O., e Stutzle, T.**, Iterated local search, Glover, F., Kochenberger, G., e Hillier, F., editors, *Handbook of Metaheuristics*, volume 57 of *International Series in Operations Research and Management Science*, páginas 320–353, Springer, 2003.
- Musa, R., Arnaout, J.-P., e Jung, H.**, Ant colony optimization algorithm to solve for the transportation problem of cross-docking network, *Computers and Industrial Engineering*, 59 (1):85 – 92, 2010, ISSN 0360–8352.
- Parragh, S. N., Doerner, K. F., e Hartl, R. F.**, A survey on pickup and delivery problems, 2008, ISSN 0344–9327.
- Rego, C. e Roucairol, C.**, *A Parallel Tabu Search Algorithm using Ejection Chains for the Vehicle Routing Problem*, Luwer Academic Publishers, Boston Massachusetts., 1996.
- Santos, F. A., Cunha, A. S., e Mateus, G. R.**, Modelos de otimização para o problema de roteamento de veículos com cross-docking, *XLII SBPO*, 2010.

Santos, F. A., Mateus, G. R., e da Cunha, A. S., A novel column generation algorithm for the vehicle routing problem with cross-docking, *Network Optimization*, volume 6701, páginas 412–425, Springer Berlin / Heidelberg, 2011a.

Santos, F. A., Mateus, G. R., e da Cunha, A. S., A branch-and-price algorithm for a vehicle routing problem with cross-docking, *Electronic Notes in Discrete Mathematics*, 37:249–254, 2011b.

Solomon, M. M., Algorithms for the vehicle routing and scheduling problems with time window constraints, *Operations Research*, 35(2):254–265, 1987.

Wen, M., Larsen, J., Clausen, J., Cordeau, J.-F., e Laporte, G., Vehicle routing with cross-docking, *Journal of the Operational Research Society*, 60:6868 – 6873, 2009, ISSN 1708–1718, doi: doi:10.1057/jors.2008.108.

Yu, W. e Egbelu, P. J., Scheduling of inbound and outbound trucks in cross docking systems with temporary storage, *European Journal of Operational Research*, 184:377–296, 2008.