

UNIVERSIDADE FEDERAL FLUMINENSE

GUSTAVO REZENDE CARVALHO

**Um Algoritmo ILS para o Problema de Localização-
Roteamento Capacitado com Dois Níveis**

Niterói
Agosto-2014

UNIVERSIDADE FEDERAL FLUMINENSE

GUSTAVO REZENDE CARVALHO

**Um Algoritmo ILS para o Problema de Localização-
Roteamento Capacitado com Dois Níveis**

Tese submetida ao Programa de Pós-Graduação em Computação da Universidade Federal Fluminense como requisito para a obtenção do título de Doutor.

Área de Concentração: Algoritmo e Otimização.

Orientador:
Prof. Luiz Satoru Ochi

Co-Orientadora:
Profa. Aline de Paula Nascimento

Niterói
Agosto-2014

Um Algoritmo ILS para o Problema de Localização-Roteamento
Capacitado com Dois Níveis

Gustavo Rezende Carvalho

Tese submetida ao Programa de Pós-Graduação em Computação da Universidade Federal Fluminense como requisito para a obtenção do título de Doutor.

Área de Concentração: Algoritmo e Otimização.

Aprovada por:

Prof. Luiz Satoru Ochi (Orientador)
Universidade Federal Fluminense (IC-UFF)

Prof^a. Aline de Paula Nascimento (Co-Orientadora)
Universidade Federal Fluminense (IC-UFF)

Prof. Lucídio dos Anjos Formiga Cabral
Universidade Federal da Paraíba (UFPB)

Prof. Luidi Gelabert Simonetti
Universidade Federal Fluminense (IC-UFF)

Prof. Nelson Maculan Filho
Universidade Federal do Rio de Janeiro (UFRJ)

Prof. Professor Yuri Abitbol De Menezes Frota
Universidade Federal Fluminense (IC-UFF)

Niterói, 8 de Agosto de 2014

Agradecimentos

Ao Prof. Dr. Luiz Satoru Ochi, pela atenção e apoio durante o processo de definição e orientação.

A Prof^a. Dr^a. Aline de Paula Nascimento, pela atenção e apoio durante o processo de definição e orientação.

Aos meus pais, Prof. Leandro Baptista Carvalho Filho e Profa. Vania Rezende Carvalho, pela atenção e apoio durante o processo de definição e orientação.

Aos meus Irmãos, Leandro Rezende Carvalho e Evandro Rezende Carvalho, pela atenção e apoio durante o processo de definição e orientação.

Ao Programa de Pós-Graduação Graduação em Informática da Universidade Federal Fluminense, pela oportunidade de realização do curso de Doutorado.

A Coordenação de Aperfeiçoamento de Nível Superior, pela concessão da bolsa de Doutorado e pelo apoio financeiro para realização desta pesquisa.

Ao Prof. Dr. Lucídio dos Anjos Formiga Cabral, pela atenção e apoio durante o processo de definição, orientação e disponibilização do Laboratório LAPORTE.

Ao Prof .Dr Anand Subramanian, pela atenção e apoio durante o processo de orientação.

A Prof^a. Dr^a. Juliana Mendes Nascente Silva, pela orientação com as bibliotecas MPI.

Ao Doutorando Hugo Harry Frederico Ribeiro Kramer pela atenção e apoio durante o processo de orientação.

A Jorge Guida Caetano dos Santos e seus Pais, pelo apoio durante o processo de orientação.

Ao Prof .Dr Gustavo Semman, pela atenção e apoio durante o processo de orientação.

Aos meus Tios João Carlos Tress, Tereza Cristina Duque-Estrada, Luiz Armando, Tania Rezende, Terezinha de Jesus pelo apoio durante o processo de orientação.

Aos meus Primos Alexandre, Claudia, João Gabriel, Dudu e Lucas pelo apoio durante o processo de orientação.

A Carlos Eduardo, pela ajuda e orientação de como utilizar o Clustar OSCAR.

Aos meus amigos, Baiano, Black Label, Carlão, Dimitri, Eyder, Flavio, Fred, Gilberto, Gruend, Gustavo, Josey, Joseph, Igor, Leandro, Marcelo, Mestre Fernando, Michael, Netto, Pablo, Paulo Gustavo, Puca, Rodrigo, Rafael, Thaís, Thiago, Toca, Wallace, Wanyne por apoiar e compartilhar os momentos alegres e tristes durante a realização do curso de Doutorado.

Aos meus Sogros Glauco Eggers e Rejane Vasconcelos, pelo apoio durante o processo de orientação.

Agradeço a minha companheira Fernanda Vasconcelos Eggers, pelo apoio e ajuda durante o processo de orientação.

Resumo

O presente trabalho trata do Problema de Localização-Roteamento Capacitado com dois níveis. Em função da elevada complexidade computacional do problema, métodos heurísticos são comumente utilizados para gerar soluções de boa qualidade em um tempo computacionalmente aceitável. Neste contexto, um procedimento baseado na meta-heurística ILS (*Iterated Local Search*) com um procedimento de busca local VND (*Variable Neighborhood Descendent*) é proposto. O algoritmo foi testado em instâncias encontradas na literatura mostrando-se capaz de encontrar melhores soluções. Além disso, versões paralelas do algoritmo foram implementadas para explorar todo o potencial do algoritmo diante da facilidade ao acesso a arquiteturas com vários núcleos de processamento.

Palavras chave: Localização, Roteamento, ILS, VND, Parallel.

Abstract

The present work deals with the Capacitated Location-Routing Problem with Two Layers. Due the high computational complexity of the problem, heuristics are commonly used to generate good quality solutions in a computationally acceptable time. In this context, a procedure based on ILS (*Iterated Local Search*) with VND (*Variable Neighborhood Descendent*) local search is proposed. The algorithm has been tested in benchmark problems found in the literature and was able to find better solutions. Furthermore, parallel versions were implemented to explore all the potential of the algorithm due to the easy accessibility to multi-core architectures.

Key words: Location, Routing, ILS, VND, Parallel.

Sumário

LISTA DE FIGURAS	VIII
LISTA DE TABELAS	IX
INTRODUÇÃO.....	10
CAPÍTULO 1 – DELIMITAÇÃO DO OBJETO	13
1.1 OBJETIVO GERAL	13
1.2 OBJETIVOS ESPECÍFICOS	13
CAPÍTULO 2 – PROBLEMA DE LOCALIZAÇÃO-ROTEAMENTO.....	15
2.1 LOCALIZAÇÃO E ROTEAMENTO	17
2.2 DEFINIÇÃO FORMAL.....	18
2.2 MODELO MATEMÁTICO	19
3.2.1 Formulação do PLRC com 2 Índices Não Orientado	20
3.2.1 Formulação do PLRC com 3 Índices Não Orientado	22
2.3 RESUMO	23
CAPÍTULO 3 – REVISÃO DA LITERATURA.....	24
3.1 MÉTODOS HEURÍSTICOS.....	24
3.2 MÉTODOS EXATOS	28
3.3 ESTUDOS DE CASO	34
3.4 ESTUDOS DE REVISÃO	39
3.5 RESUMO	39
CAPÍTULO 4 – ALGORITMO PROPOSTO	41
4.1 CONSTRUÇÃO	44
4.1.1 Inserção Paralela.....	44
4.1.2 Inserção Alternada	45
4.1.3 First Fit Decreasing.....	46
4.2 RVND	47
4.3 PERTURBAÇÕES	51
4.4 RESUMO	52
CAPÍTULO 5 – RESULTADOS	54
5.1 COMPARAÇÃO COM AS MELHORES SOLUÇÕES DA LITERATURA.....	55
5.2 COMPARAÇÃO COM O ALGORITMO ALNS	58
5.3 RESUMO	61
CAPÍTULO 6 – VERSÃO PARALELA.....	62
6.1 VERSÕES PARALELAS	63
6.1.1 Versão sem comunicação.....	64
6.1.2 Versão com Comunicação.....	65
6.2 ANÁLISE EXPERIMENTAL: $V_{DISTRIB} \times V_{PARALLEL} \times V_{PARALLEL_MPI}$	67
6.2.1 Análise Inicial: Cluster LAPORTE.....	68
6.2.2 Análise Complementar: Cluster OSCAR	82
6.3 VERSÕES PARALELAS COM REFINAMENTO DIVERSIFICADO.....	92
6.3 ANÁLISE EXPERIMENTAL - CONCLUSÕES.....	106
CONCLUSÕES E ETAPAS FUTURAS	112
REFERÊNCIAS	116
APÊNDICE.....	125

Lista de Figuras

Figura 1: Instância Cord50-5-1b com topologia de dois níveis.....	15
Figura 2: Topologia de três níveis (Barreto, 2004)	16
Figura 3: Topologia de quatro níveis (Barreto, 2004).	17
Figura 4: Gráfico do comportamento da solução no procedimento ILS (Lourenço, 2002).	41
Figura 5: Vizinhança <i>Shift 1</i> e <i>Shift 2</i>	48
Figura 6: Vizinhança <i>Swap (1,1)</i> , <i>Swap (2,1)</i> e <i>Swap (2,2)</i>	49
Figura 7: Vizinhança <i>cross</i>	50
Figura 8: Perturbações entre Depósitos.....	52
Figura 9: As três classificações proposta por Alba et al para meta-heurísticas baseadas em busca-local. (a) <i>Parallel exploration and evaluation of the neighborhood</i> , (b) <i>parallel multistart model</i> e (c) <i>parallel evaluation of a single solution</i> (Alba, 2012).	63
Figura 10: Distribuição de trabalho da versão <i>vDistrib</i>	65
Figura 11: Distribuição de trabalho e comunicação da versão <i>vParallel</i>	66
Figura 12: Distribuição de trabalho e comunicação da versão <i>vParallel_MPI</i>	67
Figura 13: Melhor Solução x <i>Threads</i> – Banco de Teste 1	72
Figura 14: <i>Speedup</i> x <i>Threads</i> – Banco de Teste 1	72
Figura 15: Melhor Solução x <i>Threads</i> – Banco de Teste 2	76
Figura 16: <i>Speedup</i> x <i>Threads</i> – Banco de Teste 2	76
Figura 17: Melhor Solução x <i>Threads</i> – Banco de Teste 3	80
Figura 18: <i>Speedup</i> x <i>Threads</i> – Banco de Teste 3	80
Figura 19: <i>Speedup</i> x <i>Threads</i> – <i>vParallel_MPI</i> – Banco de Teste 1	83
Figura 20: <i>Speedup</i> x <i>Threads</i> – <i>vParallel_MPI</i> – Banco de Teste 2	84
Figura 21: <i>Speedup</i> x <i>Threads</i> – <i>vParallel_MPI</i> – Banco de Teste 3	84
Figura 22: Melhor Solução x <i>Threads</i> – <i>vParallel_MPI</i> – Banco de Teste 1	85
Figura 23: Melhor Solução x <i>Threads</i> – <i>vParallel_MPI</i> – Banco de Teste 2	85
Figura 24: Melhor Solução x <i>Threads</i> – <i>vParallel_MPI</i> – Banco de Teste 3	86
Figura 25: Gráfico Melhor Solução – Versões <i>vParallel</i> , <i>vParallel_MPI</i> e <i>vParallel_MPI_B</i> – 32 <i>threads</i>	91
Figura 26: Gráfico <i>Speedup</i> – Versões <i>vParallel</i> , <i>vParallel_MPI</i> e <i>vParallel_MPI_B</i> – 32 <i>threads</i>	91
Figura 27: Esquema gráfico da versão paralela com refinamento diversificado.....	93
Figura 28: Gráfico Melhor Solução –Versões <i>vParallel</i> , <i>vParallel_MPI</i> , <i>vParallel_MPI_B</i> , <i>vRefina1_CD</i> , <i>vRefina2_CD</i> , <i>vRefina1_C</i> e <i>vRefina2_C</i> – 32 <i>threads</i>	102
Figura 29: Gráfico Melhor Solução – Versões <i>vParallel</i> , <i>vParallel_MPI</i> , <i>vParallel_MPI_B</i> , <i>vRefina1_CD</i> , <i>vRefina2_CD</i> , <i>vRefina1_C</i> e <i>vRefina2_C</i> – 32 <i>threads</i>	102
Figura 30: Gráfico <i>Speedup</i> - Instâncias com mais de 50 clientes – 32 <i>threads</i> – Banco de Teste 1	104
Figura 31: Gráfico <i>Speedup</i> - Instâncias com mais de 50 clientes – 32 <i>threads</i> - Banco de Teste 2	104
Figura 32: Gráfico Melhor Solução – Sem Instâncias 100-10-1a e 100-10-1b – 32 <i>threads</i> - Banco de Teste 2	105

Lista de Tabelas

Tabela 1: Tabela Cronológica de Artigos	30
Tabela 2: Estudos de Caso - Ordem Cronológica	38
Tabela 3: Estudos de Revisão – Ordem Cronológica.....	39
Tabela 4: Comparação das melhores soluções conhecidas com o ILS-RVND – Banco de Teste 1	56
Tabela 5: Comparação das melhores soluções conhecidas com o ILS-RVND – Banco de Teste 2	57
Tabela 6: Comparação das melhores soluções conhecidas com o ILS-RVND – Banco de Teste 3	58
Tabela 7: Comparação entre os algoritmos ALNS e ILS-RVND – Banco de Teste 259	
Tabela 8: Comparação entre os algoritmos ALNS e ILS-RVND – Banco de Teste 360	
Tabela 9: Comparação do sequencial com <i>vDistrib</i> – Banco de Teste 1	69
Tabela 10: Comparação do sequencial com <i>vParallel</i> – Banco de Teste 1	70
Tabela 11: Comparação do sequencial com <i>vParallel_MPI</i> – Banco de Teste 1	71
Tabela 12: Comparação do sequencial com <i>vDistrib</i> – Banco de Teste 2	73
Tabela 13: Comparação do sequencial com <i>vParallel</i> – Banco de Teste 2	74
Tabela 14: Comparação do sequencial com <i>vParallel_MPI</i> – Banco de Teste 2	75
Tabela 15: Comparação da versão sequencial com <i>vDistrib</i> – Banco de Teste 3.....	77
Tabela 16: Comparação do sequencial com <i>vParallel</i> – Banco de Teste 3	78
Tabela 17: Comparação do sequencial com <i>vParallel_MPI</i> – Banco de Teste 3	79
Tabela 18: Comparação da versão sequencial com as versões paralelas <i>vParallel_MPI</i> e <i>vParallel_MPI_B</i> – Banco de Teste 1.....	88
Tabela 19: Comparação da versão sequencial com as versões paralelas <i>vParallel_MPI</i> e <i>vParallel_MPI_B</i> – Banco de Teste 2.....	89
Tabela 20: Comparação da versão sequencial com as versões paralelas <i>vParallel_MPI</i> e <i>vParallel_MPI_B</i> – Banco de Teste 3.....	90
Tabela 21: Comparação da versão sequencial com as versões paralelas <i>vRefina1_CD</i> e <i>vRefina2_CD</i> – Banco de Teste 1.....	94
Tabela 22: Comparação da versão sequencial com as versões paralelas <i>vRefina1_CD</i> e <i>vRefina2_CD</i> – Banco de Teste 2.....	95
Tabela 23: Comparação da versão sequencial com as versões paralelas <i>vRefina1_CD</i> e <i>vRefina2_CD</i> – Banco de Teste 3.....	96
Tabela 24: Comparação da versão sequencial com as versões paralelas <i>vRefina1_C</i> e <i>vRefina2_C</i> – Banco de Teste 1	98
Tabela 25: Comparação da versão sequencial com as versões paralelas <i>vRefina1_C</i> e <i>vRefina2_C</i> – Banco de Teste 2	99
Tabela 26: Comparação da versão sequencial com as versões paralelas <i>vRefina1_C</i> e <i>vRefina2_C</i> – Banco de Teste 3	100
Tabela 27: Melhor Solução – Banco de Teste 1.....	107
Tabela 28: Melhor Solução – Banco de Teste 2.....	108
Tabela 29: Melhor Solução – Banco de Teste 3.....	109

Introdução

Devido ao grande crescimento demográfico, juntamente à falta de planejamento urbano, vários problemas comuns a vida na cidade se agravaram sendo os principais deles relacionados à distribuição de itens básicos. Por exemplo, a distribuição de água, energia e saneamento básico. Outros itens também são prejudicados: saúde, educação, lazer, coleta de resíduos, alimentação, mobilidade urbana e serviços.

Podemos destacar a mobilidade urbana como uma das principais atividades prejudicadas pelo desenvolvimento urbano não planejado, já que é através deste recurso que muitos outros serviços são providos. Tomando como exemplo as grandes cidades brasileiras, que além de possuírem estradas de má qualidade, não comportam o volume de veículos que nelas circulam. O incentivo governamental para compra de carros fez com que a frota quase dobrasse nos últimos 10 anos em prol da economia (Ribeiro, 2012), sem uma solução para o aumento do fluxo de carros. Destacamos que, no Brasil, a maioria das cargas transportadas e serviços prestados internamente no país é feito por rodovias. Consequentemente, os engarrafamentos tornaram-se rotina em quase todas as capitais brasileiras.

Várias medidas podem e devem ser tomadas para contornar a situação. Uma delas é melhorar os sistemas de distribuição. Como citado anteriormente, o gargalo para o atendimento é a mobilidade urbana. Este recurso tem que ser usado economicamente, já que todos os outros serviços dependem dele. Ao melhorar o sistema de distribuição de um serviço, fazendo que sua frota permaneça menos tempo rodando, a população e os outros prestadores de serviço são beneficiados e menos veículos passam a rodar nas vias públicas. Como os veículos utilizados pelos prestadores de veículos, em geral, são de grande porte, ocupando grande espaço das vias e trafegam em baixa velocidade, a presença de menor quantidade deles tornará o tráfego mais ágil, aliviando o fluxo.

Em alguns casos, um tipo de serviço pode ainda não existir em certa localidade. Decidir onde instalar pode trazer benefícios ou transtornos à população. Um exemplo de serviço que pode ser mal localizado são os centros de coleta de resíduos, que desvalorizam a área ao seu redor e trazem perigo e transtornos à

população. No entanto, precisam ter uma localização que facilite a coleta. Outro tipo de decisão a ser considerada na prestação de serviços é a decisão de abrir mais de um centro de distribuição ou coleta para reduzir custos e facilitar a prestação de um serviço.

O problema de localização-roteamento (PLR) é definido em um cenário onde se deseja localizar centros de distribuição/coleta (depósitos) associados a um conjunto de clientes que deverão ser atendidos por suas devidas demandas de produtos ou serviços por intermédio de um veículo, gerando uma rota. Vários estudos de casos foram realizados trazendo benefícios a população local (Jacobsen e Madsen, 1980). Os problemas estudados foram coleta de resíduos, distribuição de sangue, coleta de resíduo nucleares, entre outros. Existem várias organizações hierárquicas estudadas para este tipo de problema, porém a mais abordada na literatura é a de dois níveis (Maranza, 1963; Chrisotphides e Eilon, 1969; Laporte, 1981, 1983, 1986, 1988, 1989; Perl e Daskin, 1984; Tuzun e Burk 1999; Wu, 2002; Cappanera, 2004; Prins, 2004, 2006, 2007; Barreto, 2004, 2007; Duhamel, 2008; Akca, 2009; Contardo, 2010; Belenguer, 2010, 2011; Baldacci, 2011; Hemmelmayr, 2011). Em um sistema de localização com três níveis, situam-se no primeiro nível os serviços primários habitualmente composto por unidades produtivas (hospitais, centros de coleta de lixo, aeroportos etc). No segundo nível encontram-se os serviços secundários, que representam locais de atividades intermediárias tais como armazéns, centros de distribuição, terminais de contentores etc. O terceiro nível é associado aos clientes.

Apesar de alguns estudos terem sido realizados no PLR com três níveis, este caso em particular se torna muito abstrato quando estamos lidando com fábricas no cenário brasileiro. Diversas variáveis tem que ser levadas em consideração, como incentivo fiscal, ICMS, IPTU etc, tornando difícil prever o custo de instalação. Muitas vezes o incentivo estadual torna a decisão de instalação da nova fábrica uma negociação entre estados. Além disso, o problema de três níveis pode ser modelado como dois sub-problemas de dois níveis. Outro ponto a ser considerado é que, se houverem poucas localidades para a abertura de uma fábrica, pode-se gerar uma instância para cada caso e a decisão de localizar só será utilizada para os centros de distribuição no segundo nível.

A contribuição principal deste trabalho é estudar e propor uma solução heurística para o PLR de dois níveis capacitado, o primeiro nível pode ser composto

por unidades produtivas ou serviços secundários e no segundo nível estão os clientes. Os centros que distribuirão produtos ou serviços serão chamados de depósitos. Além disso, o PLR é um problema de otimização de alta complexidade, sendo sua versão de decisão definida como um problema NP-Completo por Barreto (2004). Como consequência, é exigido um grande esforço computacional para encontrar uma solução ótima à medida que as instâncias aumentam de tamanho. Duas abordagens são comumente utilizadas: as exatas e as heurísticas. Na literatura, nenhum método exato resolveu problemas com mais de 100 nós sem utilizar grandes simplificações do problema. Em contrapartida, as meta-heurísticas ganham espaço e produzem soluções de boa qualidade em um tempo computacional aceitável para o PLR.

Vários métodos tem sido propostos para paralelizar meta-heurísticas para os mais diversos problemas de otimização (Alba, 2012). A paralelização toma vantagem do grande poder de processamento disponível e ajuda os algoritmos a explorarem melhor o espaço de busca em um tempo aceitável. Outro ponto importante é a facilidade de acesso a arquiteturas com processadores com vários núcleos (*multicores*) e também multiprocessadores (Geer, 2005). Diante deste cenário, este trabalho propõe versões paralelas do algoritmo proposto.

O presente trabalho está organizado da seguinte forma: o Capítulo 1 apresenta os objetivos gerais e específicos da proposta; o Capítulo 2 descreve o Problema de Localização-Roteamento; o Capítulo 3 apresenta uma revisão dos trabalhos publicados no tema estudado; o Capítulo 4 apresenta o algoritmo ILS-RVND proposto para a resolução do PLR com dois níveis; o Capítulo 5 apresenta os resultados parciais obtidos em três bancos de testes propostos pela literatura; o Capítulo 6 apresenta versões paralelas da do algoritmo ILS-RVND e seus resultados; e encerra com as conclusões e sugestão das etapas futuras desta tese.

Capítulo 1 – Delimitação do Objeto

O presente trabalho trata do Problema de Localização-Roteamento com Capacidade (PLRC) de dois níveis, considerando o primeiro nível composto pelos depósitos e o segundo, pelos clientes. com o objetivo de reduzir o custo global. Uma heurística foi proposta e implementada visando solucionar o problema com o objetivo de reduzir o custo global. O custo global é composto pelo somatório dos custos de roteamento, localização dos depósitos e frota utilizada. Versões paralelas também serão desenvolvidas com base na meta-heurística implementada.

1.1 Objetivo Geral

Estudar o Problema de Localização-Roteamento identificando quais heurísticas produzem soluções de boa qualidade, através da revisão de métodos já implementados e na busca de versões mais aprimoradas, propondo versões paralelas em vista de novas arquiteturas de processadores, advindas da disponibilidade de vários núcleos e a maior capacidade de processamento.

1.2 Objetivos Específicos

Levantar o estudo da arte sobre os métodos utilizados para solucionar o Problema de Localização-Roteamento, na perspectiva de efetuar o estudo dos algoritmos e seus resultados, observando vantagens e desvantagens, como também da modelagem necessária para cada tipo de variante e instâncias utilizadas nos testes.

Desenvolver um algoritmo baseado na meta-heurística *Iterated Local Search* (ILS) com a fase de busca local baseada em *Variable Neighborhood Decent* (VND), estudando a influência dos critérios utilizados durante a fase de construção e determinando sua influência no custo global. Na fase de busca local, selecionar um conjunto de algoritmos que refinem as soluções obtidas na fase de construção e um conjunto de métodos de perturbação que permitam sair de mínimos locais de baixa qualidade.

Produzir uma versão paralela do algoritmo sequencial com foco nas arquiteturas de processadores com vários núcleos através de *threads* para utilizar todo o poder de processamento já existente na maioria das máquinas atuais. Em seguida, realizar uma versão paralela hierárquica com a *Message Passing Interface* (MPI) (Message Passing Forum, 1995) para ser utilizada em *clusters* com o objetivo de tirar proveito de uma plataforma com mais processadores, onde é possível aumentar o processamento em busca de uma melhor solução sem aumentar o tempo de resposta para o usuário ou distribuir o processamento em busca de um menor tempo de resposta (Dongarra, 2003; David Geer, 2005; Intel, 2006).

Capítulo 2 – Problema de Localização-Roteamento

O problema de localização-roteamento com restrições de capacidade (PLRC) se define pela decisão de localizar centros de distribuição/coleta (depósitos) associados a um conjunto de clientes que deverão ter suas demandas atendidas por veículos, respeitando a capacidade destes e dos depósitos (Jacobsen e Madsen, 1980). Os veículos devem retornar aos depósitos onde iniciam suas rotas. Com isso, pode-se imaginar vários cenários reais aplicados a este problema, como a localização de fábricas, centros de coleta de lixo, prestadores de serviços, lojas, aplicações militares, exploração espacial etc. Existem várias organizações hierárquicas para este tipo de problema, porém a mais abordada na literatura é a de dois níveis (Maranza, 1963; Chrisotphides e Eilon, 1969; Laporte, 1981, 1983, 1986, 1988, 1989; Perl e Daskin, 1984; Tuzun e Burk 1999; Wu, 2002; Cappanera. 2004; Prins, 2004, 2006, 2007; Barreto, 2004, 2007; Duhamel, 2008; Akca, 2009; Contardo, 2010; Belenguer, 2010, 2011; Baldacci, 2011; Hemmelmayr, 2011), onde teremos os centros de distribuição e os clientes. Pode-se observar uma topologia de dois níveis na Figura 1. Os depósitos são os quadrados e os clientes são os pontos pretos.

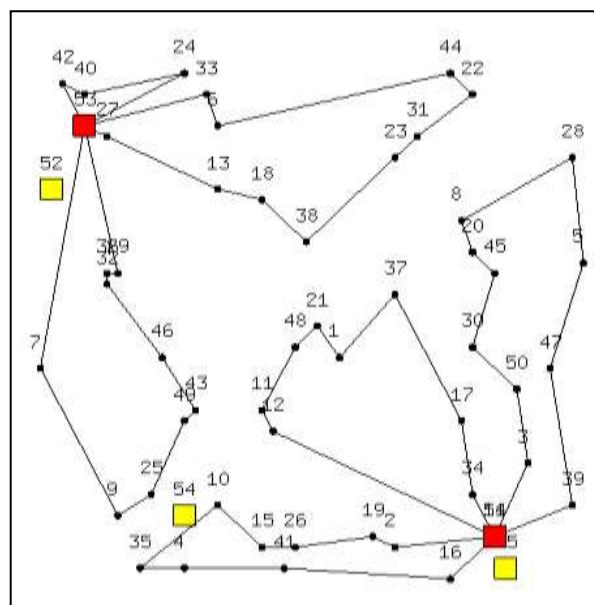


Figura 1: Instância Cord50-5-1b com topologia de dois níveis.

Em Laporte (1988) é descrito um sistema de localização composto por três níveis. No primeiro nível situam-se os serviços primários, habitualmente compostos por unidades produtivas (hospitais, centros de coleta de lixo, aeroportos etc). No segundo nível encontram-se os serviços secundários, que representam locais de atividades intermediárias tais como armazéns, centros de distribuição, terminais de contentores etc, designados neste texto por depósitos. O terceiro nível é associado aos clientes. Neste nível ocorre a entrega ou coleta de um produto ou a prestação de um serviço (MIN, 1996). O esquema gráfico da topologia de três níveis pode ser observado na Figura 2.

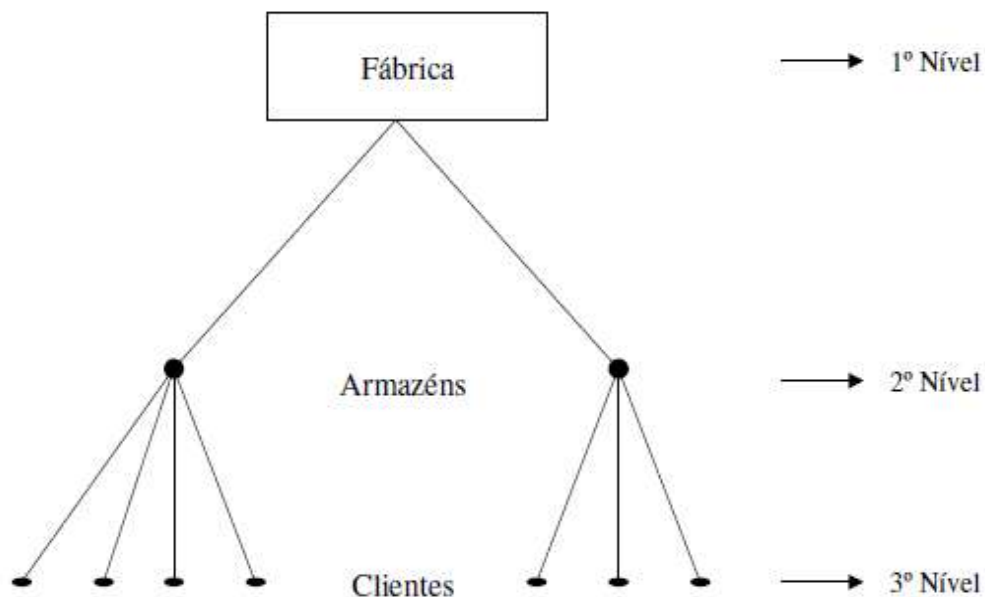


Figura 2: Topologia de três níveis (Barreto, 2004)

Outros trabalhos consideraram outras quantidades de níveis. Em Nambiar et al. (1981) encontra-se um sistema composto por quatro níveis: produtores, estações de coleta, fábricas e porto. Mercer et al. (1978) e Dejax (1988) também consideram sistemas de 4 níveis. Van Roy considera um sistema de seis níveis relacionado com produção e distribuição de combustíveis. Assim como neste trabalho, OR e Pierskal (1979) e vários outros (Gaskell, 1967, Christofides e Eilon, 1969, Min, 1992, Daskin 1995, Duhamel, 2008, Contardo, 2010, Belenguer, 2010, Baldacci, 2011, Hemmelmayr, 2011) trabalhos consideram um sistema de dois níveis. Um gráfico de um sistema de quatro níveis pode ser observado na Figura 3.

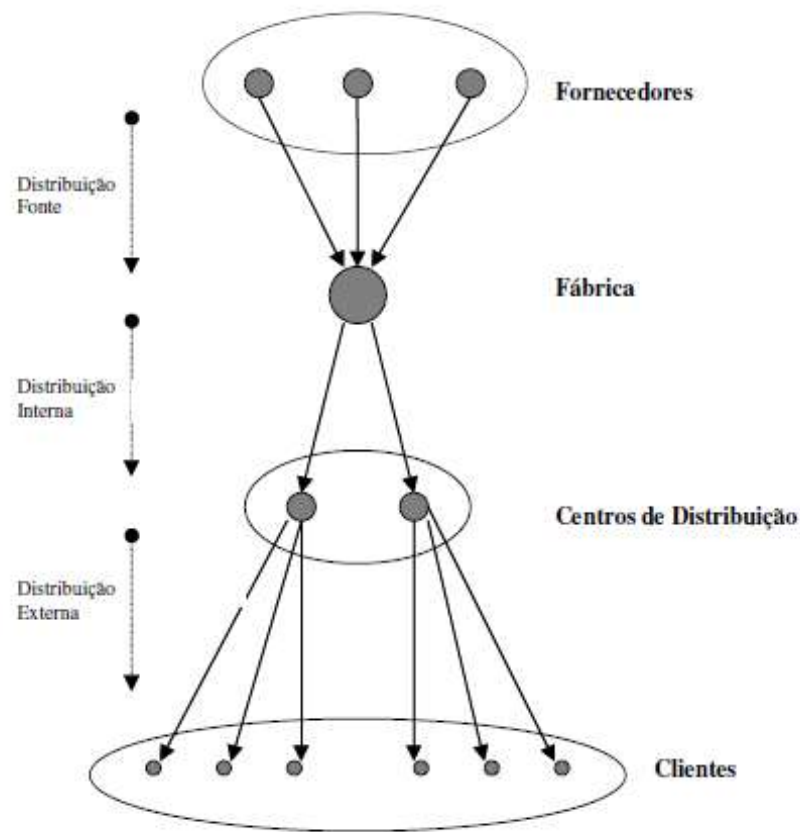


Figura 3: Topologia de quatro níveis (Barreto, 2004).

2.1 Localização e Roteamento

Pelo fato do PLRC ser definido muitas vezes como a combinação do problema de localização de facilidades com o problema de roteamento de veículos (Srivastava, 1990), não é aconselhável definir a localização dos depósitos sem levar em consideração a otimização das rotas associadas. A simples escolha de uma boa localização dos depósitos não acarreta, necessariamente, em uma diminuição dos custos globais envolvidos. Podemos citar vários casos onde uma escolha aparentemente boa de localização não acarreta em uma boa solução:

- Escolher um depósito de custo de instalação baixo, porém mal localizado acarretando em altos custos de roteamento;
- Utilizar um número de depósitos inferior ao ideal fazendo com que o custo para rotear alguns clientes não compense a economia feita na escolha dos depósitos

- Utilizar um número de depósitos superior ao ideal fazendo com que o gasto com depósitos não compense a facilidade de rotear as rotas com um maior número de depósitos.
- Levar em consideração apenas a boa localização do depósito e desconsiderar o custo dos depósitos em relação à sua proporção ao custo global (relação custo dos depósitos versus custo de roteamento).

Estas e outras premissas são defendidas por vários autores (Webb, 1968; Sussams, 1969; Marks e Stricter, 1971; Eilon, 1971; Rand, 1976; Geoffrion e Powers, 1980; Perl e Daskin, 1985; Laporte, 1988; Wu, 2002) e, em destaque o trabalho de Salhi e Rand (1989) sobre o impacto de ignorar as rotas durante a localização dos depósitos.

Existem estudos com dois tipos de localização de depósitos: contínua ou discreta. Na localização contínua, o depósito pode ser instalado em qualquer ponto do espaço de uma região preliminarmente delimitada (Weber, 1929; Cooper, 1963; Haley, 1963; Eilon, 1971; Fisher, 1981; Drezner, 1985). Este tipo de localização não é aconselhável devido ao conjunto de regras que construções do tipo de centros de distribuição têm que respeitar por questão de segurança, viabilidade de construção (normas e regras de construção impostas pela prefeituras locais), e viabilidade para os veículos entrarem e saírem da instalação. Na localização discreta, é considerado um conjunto discreto de possíveis pontos de instalação que seguirão requisitos que possibilitem as instalações (Baumol e Wolfe, 1958; Hakimi, 1964; Maranzana, 1964; Francis, 1983; Dejax, 1988; Drezner, 1995).

2.2 Definição Formal

O PLRC pode ser definido na estrutura de um grafo. Sendo $G=(V,E)$ um grafo onde V é formado pela união dos subconjuntos de vértices C e D , tal que $C=\{0,...,n\}$ representam os clientes e $D=\{n+1,...,n+d\}$ são os depósitos $k \in D$ de capacidade w_k e custo de ativação d_k . Cada aresta $(i,j) \in E$ possui custo não negativo c_{ij} e cada cliente k possui uma demanda determinística associada $q_k > 0$. Sendo $F=\{1,...,v\}$ o conjunto de veículos homogêneos de capacidade Q e custo R . O PLRC consiste em localizar (selecionar) um conjunto variável de depósitos e um

conjunto de rotas de maneira que: (i) todas as demandas devem ser satisfeitas; (ii) respeite as capacidades dos veículos e dos depósitos; (iii) o cliente seja visitado por apenas um veículo e apenas uma vez; (iv) o veículo retorne ao mesmo depósito de partida da rota; (v) minimize os custos de localização, da frota e roteamento envolvidos.

Os algoritmos neste trabalho consideram, portanto, o problema de localização-roteamento de dois níveis, com frota homogênea, capacitada, com e sem custo de utilização. Depósitos com custos e capacidade limitada e ilimitada. O objetivo é atender todas as demandas dos clientes minimizando o custo global que consiste na soma dos custos da frota utilizada, dos depósitos e de roteamento.

2.2 Modelo Matemático

A seguir apresentam-se dois modelos matemáticos propostos por Barreto (2004) para o PLRC. Um com 2 índices e outro com 3 índices. As formulações com 2 índices são concisas e envolvem um reduzido número de variáveis. Como consequência, não é possível identificar o percurso e os clientes de uma determinada rota, só sendo aplicáveis nos casos com frotas homogêneas.

As formulações com 3 índices são mais completos, em questão de informações, facilitando a intervenção sobre as características de cada rota. Como desvantagem, possui um número muito grande de variáveis e como consequência, necessitando de um grande custo computacional para a sua resolução (Barreto, 2004).

Apesar de formulações distintas para o mesmo problema, os dados são sempre os mesmos e são definidos a seguir.

$N = \{1, 2, \dots, n\}$, representa n clientes no plano,

$P = \{n + 1, n + 2, \dots, n + p\}$, representa p potenciais depósitos,

$R = \{1, 2, \dots, r\}$, representa r potenciais rotas ou veículos (formulação com 3 índices),

$E = \{e_1, e_2, \dots, e_n\}$, representa as demandas dos clientes,

$U = \{u_1, u_2, \dots, u_p\}$, representa a capacidade dos depósitos ,

w = capacidade dos veículos,

c_{ij} = custo associado a aresta que liga os clientes $i, j \in N$,

c_{ik} = custo associado a aresta que liga o cliente $i \in N$ e o depósito $k \in P$,

f_k = custo associado à instalação do depósito $k \in P$

3.2.1 Formulação do PLRC com 2 Índices Não Orientado

Definição das variáveis de decisão.

x_{ij} = número de vezes que é usado a ligação entre o cliente i e o cliente j

$y_k = \begin{cases} 1 \\ 0 \end{cases}$, 1 se o depósito for instalado em k , 0 se caso contrário.

Definição das variáveis auxiliares.

z_k = número de rotas (veículos) associados ao depósitos k .

$$\text{Min } z = \sum_{i,j \in N, i < j} c_{ij} x_{ij} + \sum_{i \in N, k \in P} c_{ik} x_{ik} + \sum_{k \in P} f_k y_k$$

Sujeito a:

$$\sum_{i \in N, i < g} x_{ig} + \sum_{j \in N, g < j} x_{gj} + \sum_{k \in P} x_{gk} = 2, \quad (g \in N) \quad (3.1)$$

$$z_k \leq \frac{u_k}{w} y_k, \quad (k \in P) \quad (3.2)$$

$$\sum_{i \in N} x_{ik} = 2z_k, \quad (k \in P) \quad (3.3)$$

$$\sum_{k \in P} z_k \geq \frac{\sum_{i \in N} e_i}{w} \quad (3.4)$$

$$x_{i_1 i_2} + 3x_{i_2 i_3} + x_{i_3 i_4} \leq 4, \quad (i_1, i_4 \in P; i_2, i_3 \in N) \quad (3.5)$$

$$x_{i_1 i_2} + 2 \sum_{i,j \in \{i_2, \dots, i_{h-1}\}} x_{ij} + x_{i_{h-1} i_h} \leq 2h - 5, \quad (h \geq 5; i_1, i_h \in P; i_2, \dots, i_{h-1} \in N) \quad (3.6)$$

$$\sum_{i,j \in S, i < j} x_{ij} \leq + |S| - \left\lceil \frac{\sum_{i \in S} e_i}{w} \right\rceil, \quad (S \subset N, 3 \leq |S| \leq |N| - 3) \quad (3.7)$$

$$\begin{aligned}
x_{ij} &\in \{0,1\} & , (i,j \in N, i < j) \\
x_{ik} &\in \{0,1,2\} & , (i \in N, k \in P) \\
y_k &\in \{0,1\} & , (k \in P) \\
z_k &\in Z_0^+ & , (k \in P)
\end{aligned} \tag{3,8}$$

A função objetivo representa o custo total do PLRC, traduzido pela soma do custo de roteamento com o custo do depósito.

As restrições (3.1), em face da binariedade das variáveis de roteamento, determinam que o grau de cada cliente é 2, ou seja, é visitado uma única vez.

Sempre que um depósito é visitado por um veículo, as restrições (3.2) selecionam este depósito e ao mesmo tempo evitam que o limite de capacidade não seja excedido. Observa-se que $\frac{u}{w}$ indica o número máximo de rotas (veículos) que pode ser suportado pelo depósito k . Nas formulações com 2 índices não é possível relacioná-los com os respectivos depósitos. Não sabendo quais os clientes relacionados a um determinado depósito, torna-se difícil cumprir as restrições de capacidade em termos da demanda dos clientes. Em vez disso, é limitado o número de veículos (rotas) a um depósito.

Devido a integralidade das variáveis z , as restrições (3.3) atribuem grau par a cada depósito, isto significa que o número de veículos que chegam nos depósitos k é igual ao número de veículos que saem.

As restrições (3.2) e (3.3) estabelecem a ligação entre as variáveis de localização e roteamento.

As restrições (3.4) estabelecem o número mínimo de rotas associada a cada depósito.

As restrições (3.5) e (3.6), propostas por Laporte et al. (1986), impedem a formação de cadeias com 2 clientes e com mais de 2 clientes entre depósitos.

O objetivo da restrição (3.7) é impedir a formação de rotas sem depósitos ou com encomendas superiores a w .

As restrições (3.8) estabelecem a binariedade ou integralidade das variáveis.

O número de variáveis nesta formulação é dado por $\frac{1}{2}n(n-1) + np + 2p$.

3.2.1 Formulação do PLRC com 3 Índices Não Orientado

Definição das variáveis de decisão.

x_{ijl} = número de vezes que é usado a ligação entre o cliente i e o cliente j na rota l ,

x_{ikl} = número de vezes que é usada a ligação entre o cliente i e o depósito k na rota l ,

$$y_k = \begin{cases} 1 \\ 0 \end{cases}, \text{ 1 se o depósito for instalado em } k, \text{ 0 se caso contrário.}$$

Definição das variáveis auxiliares.

$$t_{il} = \begin{cases} 1 \\ 0 \end{cases}, \text{ 1 se o cliente } i \text{ é servido pelo veículo } l, \text{ 0 se caso contrário.}$$

$$z_{kl} = \begin{cases} 1 \\ 0 \end{cases}, \text{ 1 se o depósito } k \text{ suporta o veículo } l, \text{ 0 se caso contrário.}$$

$$\text{Min } z = \sum_{i,j \in N, i < j, l \in R} c_{ij} x_{ijl} + \sum_{i \in N, k \in P, l \in R} c_{ik} x_{ikl} + \sum_{k \in P} f_k y_k$$

Sujeito a:

$$\sum_{l \in R} t_{il} = 1, \quad (i \in N) \quad (3.9)$$

$$\sum_{i \in N, i < g} x_{igl} + \sum_{j \in N, g < j} x_{gjl} + \sum_{k \in P} x_{gkl} = 2t_{gl}, \quad (g \in N; l \in R) \quad (3.10)$$

$$\sum_{l \in R} z_{kl} \leq \frac{u_k}{w} y_k, \quad (k \in P) \quad (3.11)$$

$$\sum_{i \in N} x_{ikl} = 2z_{kl}, \quad (k \in P; l \in R) \quad (3.12)$$

$$\sum_{k \in P, l \in R} z_{kl} \geq \frac{\sum_{i \in N} e_i}{w} \quad (3.13)$$

$$\sum_{i \in N} e_i t_{il} \leq w, \quad (l \in R) \quad (3.14)$$

$$\sum_{i,j \in S, i < j, l \in R} x_{ijl} \leq |S| - 1, \quad (S \subset N, 3 \leq |S| \leq |N| - 3) \quad (3.15)$$

$$\begin{aligned}
x_{ijl} &\in \{0,1\} & , (i, j \in N, i < j; l \in R) \\
x_{ikl} &\geq 0 & , (i \in N; k \in P; l \in R) \\
y_k &\in \{0,1\} & , (k \in P) \\
t_{il} &\in \{0,1\} & , (i \in N; l \in R) \\
z_{kl} &\in \{0,1\} & , (k \in P; l \in R)
\end{aligned} \tag{3.16}$$

As restrições (3.9) garantem que cada cliente pertence a uma rota e é servido por apenas um veículo.

As restrições (3.10) estabelecem o grau 2 para cada cliente na sua rota. As restrições (3.9) e (3.10) obrigam os clientes alocados para determinado veículo a estarem em uma mesma rota.

Quando um veículo é associado a um depósito, as restrições (3.11) impõem a sua abertura e evitam que a capacidade seja violada.

As restrições (3.12) garantem a associação de cada rota a um depósito por dois arcos, evitando rotas entre depósitos.

As restrições (3.13) impõem um número mínimo de veículos.

As sub-rotas são eliminadas através das restrições (3.15).

A binariedade das variáveis x_{ikl} é assegurada pelas restrições (3.9), (3.10), (3.12) e pela binariedade das variáveis restantes.

O número de variáveis é dado por $\left(\frac{n^2}{2} + \frac{n}{2} + np + p\right)r + p$.

2.3 Resumo

Neste capítulo, definimos o PLRC de dois níveis capacitado formalmente e se discute as dificuldades para resolvê-lo. Além disso, duas modelagens para o PLRC são apresentadas, uma com dois índices não orientado e uma com três índices e também não orientado. A seguir, o Capítulo 3 se dedica a fazer uma revisão dos principais trabalhos publicados no PLR fazendo um resumo de estudos de casos, instâncias propostas e estudos de revisão.

Capítulo 3 – Revisão da Literatura

Barreto (2004), em sua tese de doutorado, apresenta uma revisão completa dos artigos publicados em periódicos, livros e trabalhos disponíveis na internet entre os anos de 1963 e 2004. Nesta seção será feita uma atualização desta revisão bibliográfica focada nos métodos propostos na literatura para resolver o Problema de Localização-Roteamento e suas instâncias.

3.1 Métodos Heurísticos

Na história do PLR, o primeiro artigo de conhecimento do autor que apresenta uma discussão sobre o problema é o trabalho de Maranza (1963). O trabalho considera percursos entre os clientes e os depósitos não necessariamente diretos e distâncias ponderadas pelo volume transportado. A técnica proposta é heurística, onde a localização dos depósitos baseia-se na determinação dos centros de gravidade entre os nós (clientes) e o critério de caminho mais curto entre os clientes e depósitos para determinar a distribuição. Para esta decisão do caminho mais curto foi utilizado o algoritmo de Bellman-Ford (1958). Os testes foram realizados em um problema real que utiliza 40 cidades Italianas, que neste caso são consideradas os clientes, e três depósitos. Em Griffiths (1968), apresenta-se um problema de localização e roteamento que possui três níveis. Neste trabalho o custo da rota ainda é calculado através da distância euclidiana entre os clientes e os depósitos em vez de considerar o percurso realizado entre cada cliente em cada rota.

Um passo importante é dado por Webb (1968) onde se estabelece uma relação entre a soma do custo da rota, calculado através do algoritmo de Clarke e Wright (Clarke, 1964), e a soma das distâncias dos clientes aos depósitos. Conclui-se que não é apropriado utilizar a soma entre as distâncias dos clientes e os depósitos em virtude da discrepância de resultados quando comparado com o método que soma o custo da rota. Além disso este trabalho apresenta clientes com demandas e veículos com capacidade definida.

Christofides e Eilon (1969) publicam um estudo sobre o problema de roteamento de veículos a partir de um depósito. O problema é estudado em cima de uma configuração com n clientes aleatória e uniformemente distribuídos no plano e m veículos a serem posicionados. Neste artigo é verificado que quando há um grande número de clientes em uma área ou o número de clientes por rota é pequeno, a localização dos depósitos não varia muito.

O trabalho de Perl e Daskin (1984) apresenta um PLR com demandas por clientes e custos por instalação dos depósitos, custo variável por depósito e restrição de capacidade para veículos e depósitos. É estabelecido um número máximo de veículos. Os custos de distribuição são euclidianos. O objetivo é determinar os depósitos a serem abertos e rotear os clientes minimizando o custo global. O problema é solucionado através de três fases. Na primeira, resolve-se o roteamento dos clientes considerando todos os depósitos em aberto e respeitando a capacidade dos veículos. Em seguida, resolve-se otimamente o Problema Localização-Atribuição usando enumeração implícita. Os depósitos que são abertos na fase dois são usados na terceira fase em que cada um dos clientes é reatribuído a um dos depósitos em aberto e são determinadas as melhores rotas dos depósitos aos seus clientes. As duas últimas fases são realizadas iterativamente até não haver melhorias no custo global. Em 1985, o trabalho foi revisado pelos autores e novos testes foram realizados em cima de instâncias maiores. Também apresentam um estudo em uma aplicação à Indústria Americana utilizando apenas dois depósitos.

Como foi citado anteriormente na seção 2.1, vários trabalhos falam sobre o efeito de ignorar o roteamento ao localizar os depósitos, mas, em destaque, temos trabalho de Salhi e Rand (1989) que se dedica inteiramente sobre este impacto. O trabalho consiste em aplicar vários algoritmos para determinar a melhor localização dos depósitos com base na distância radial para, em seguida, gerarem rotas para cada uma das localizações determinadas. Os autores chegam à conclusão de que, em alguns casos, a pior localização permite obter a melhor solução após a obtenção das rotas.

Outro trabalho em destaque é o de Srivastava e Benton (1990) que trata da disposição física do PLR testando a sensibilidade a determinados fatores, como a estrutura de custos, a distribuição espacial dos clientes e o número de potenciais depósitos. Clientes uniformemente distribuídos versus clientes aglomerados é um fator que não difere o resultado final. Baixos custos de localização e altos custos de

distribuição podem beneficiar um grande número de depósitos. No caso dos custos de localização serem superiores aos de distribuição, o número de depósitos será reduzido. As instâncias para testes foram geradas aleatoriamente com uma distribuição uniforme. Algumas instâncias consideram os clientes espacialmente agrupados em áreas restritas e aleatoriamente distribuídos dentro destas áreas uniformemente. As demandas dos clientes foram geradas aleatoriamente com uma distribuição uniforme. Os veículos possuem capacidade fixa. São testadas três heurísticas: *Saving-Drop*, *Saving Add* e *Cluster-Routing*. O algoritmo *Saving-Drop* mostrou-se melhor que os outros dois métodos por não ser sensível ao tipo de instância.

Em 1993, Srivastava propõe três métodos heurísticos para a resolução do PLR: SAV1, SAV2 e CLUST. O algoritmo SAV1 utiliza todos os depósitos disponíveis e usa o custo aproximado das rotas para determinar quais depósitos deixarão de ser utilizados. O custo das rotas é estimado através de uma versão do algoritmo *saving* para múltiplos depósitos. O refinamento da solução é feito iterativamente através das fases de roteamento e fechamento de depósitos. A versão SAV2 parte da adição de depósitos, um de cada vez. O algoritmo CLUST é baseado em agrupamento que utiliza a técnica de pesquisa de densidade. Como resultado, o algoritmo SAV1 se mostrou mais indicado para dados uniformemente distribuídos e o CLUST para dados que possuem a característica de clientes agrupados.

Os autores Salhi e Fraser (1996) introduziram novas variáveis ao PLR, como frota heterogênea, custos fixos e variáveis nos veículos, aproximação explícita do comprimento das rotas e resolução de problemas de maiores dimensões. O objetivo neste trabalho é determinar o número e localização dos depósitos, as rotas dos veículos e a frota a ser utilizada. O algoritmo utilizado possui duas fases: a primeira é responsável por localizar os depósitos e atribuir os clientes; a segunda fase é responsável pela escolha da frota e o roteamento dos clientes. No mesmo ano deste trabalho, Nagy e Salhi publicam dois artigos sobre um método chamado *nested* para a resolução de um PLR com demanda por cliente, capacidade por veículo e distância máxima para cada rota. O novo método, comparado ao publicado no mesmo ano, adiciona a distribuição no procedimento de localização. Além disso, o algoritmo utiliza busca tabu e considera três tipos de vizinhança: *add*, *drop* e *shift*.

Um PLR de três níveis com restrições de capacidade nos veículos e depósitos é abordado no trabalho de Min (1996). Utiliza-se um método heurístico dividido em três etapas: agregação dos clientes mais próximos em grupos que cumpram as restrições de capacidade (Ward, 1963); localização dos depósitos com atribuição dos fornecedores e clientes ao depósito mais próximo; e configuração individual das rotas.

Os autores Tuzun e Burke (1999) propõem uma heurística baseada em busca tabu para resolver um PLR de dois níveis capacitado, formulado em programação linear. A heurística possui duas fases: na primeira fase, a busca tabu é realizada em cima das localizações; a segunda fase trabalha em cima da atribuição dos clientes. A pesquisa é realizada iterativamente entre as duas fases até não obter mais melhoras no custo global. Sementes aleatórias são utilizadas no início do algoritmo para escolher os primeiros depósitos. O algoritmo *saving*, proposto por Clarke e Wright (1964) para o clássico problema de roteamento de veículos, é aplicado para o roteamento e utiliza-se o mecanismo de busca local *2-optimal*. Neste trabalho é proposto o banco de teste conhecido como “TUZUN”. O banco de teste possui 36 instâncias com 100 a 200 clientes e 10 a 20 possíveis depósitos. Os depósitos não possuem limite de capacidade.

Os modelos PLR apresentados anteriormente muitas vezes consideram número ilimitado de veículos para cada depósito. Em contraposição, o trabalho de Wu et al. (2002) propõe um PLR com múltiplos depósitos, número limitado de veículos em cada depósito e frota heterogênea, além de uma resolução heurística baseada em *simulated annealing* onde as fases de localização e roteamento interagem.

Várias abordagens baseadas em agrupamento são propostas por Barreto et al. (2007) para solucionar o PRL capacitado de dois níveis. Os testes são realizados em cima do banco de teste disponibilizado pelo autor em 2003. O banco de teste é composto por 19 instâncias, algumas obtidas da literatura (Or, 1976, Perl, 1983) e outras adaptadas de instâncias relacionadas com o PRV (Gaskell, 1967, Christofides e Eilon, 1969, Min et AL., 1992, Daskin, 1995). É importante observar que este é o único autor que realiza testes em duas das instâncias com 318 clientes e 4 possíveis localizações de depósitos. Os autores citados que realizaram testes neste banco não divulgam nenhuma solução para estas duas instâncias.

Uma abordagem memética é proposta por Duhamel et al. (2008). Os testes são realizados em uma seleção de instâncias dos bancos de testes “BARRETO”, “PRINS” e “TUZUN”. Nenhum resultado de instância com mais de 100 clientes foi publicado.

Um relatório técnico publicado por Hemmelmayr et al. (2011) propõe uma heurística adaptativa de busca em vizinhança. O algoritmo utiliza métodos parecidos com perturbações e buscas locais com penalidades para guiar estes movimentos. É a heurística que obteve, até então, os melhores resultados para os bancos de testes “PRIN” e “TUZUN”.

3.2 Métodos Exatos

Uma série de artigos foram publicados na década de oitenta por Laporte et al (1981, 1983, 1986, 1988, 1989). Em 1981, os autores apresentam uma resolução exata para um PLR com um único depósito. É utilizada uma formulação de programação linear inteira para o problema considerando a localização de apenas um depósito e m percursos de caixeiro viajante. A função objetivo é formada pelo somatório no custo fixo de localização dos depósitos com os custos de distribuição. A resolução se dá por um algoritmo de *branch-and-cut* em que as restrições de eliminação de sub-ciclos são introduzidas à medida que se verifica sua violação. No trabalho publicado em 1983, os autores consideram três tipos de problemas: problemas euclidianos em que o número de veículos por depósito não é especificado; problemas não euclidianos em que o número de veículos por depósito não é definido; e problemas em que são permitidas várias visitas a um mesmo cliente. Os métodos de resolução são variantes do mesmo método utilizado no artigo anterior.

O problema abordado em 1986 por Laporte et al. é um PLRC com número determinado de veículos por depósito, demanda por cliente e restrições de capacidade para os veículos. Novamente utilizam *branch-and-cut* para solucionar o problema, mas incorporam no algoritmo as restrições de capacidade de veículos nas restrições de eliminação de sub-ciclos. No artigo subsequente, apresentam um PLRC assimétrico. Neste PLRC se admite um conjunto de potenciais depósitos com custo de operação, demanda para clientes, capacidade para os veículos e custos fixos iguais. O número de veículos em cada depósito não pode exceder um limite

pré-definido e o custo das rotas não pode exceder um determinado limite. Para resolver o problema proposto pelos autores em 1988, uma modificação na representação do grafo é proposta resultando em uma reformulação do problema como um problema de atribuição, que é resolvido utilizando o método de *branch-and-bound*.

O trabalho de Perl e Daskin em 1989 utiliza novamente um *branch-and-bound* para resolver um PLR com encomendas aleatórias dos clientes. O estudo é separado em dois problemas de acordo com as restrições probabilísticas. No primeiro caso, a probabilidade de falhar pelo menos uma vez uma rota planejada não pode exceder um valor α . No segundo problema, a penalidade esperada em uma rota não pode exceder uma fração β do seu comprimento planejado. O objetivo do problema é controlar a frequência de falhas nas rotas.

O trabalho de Cappanera et al. (2004) lida genericamente com o problema de localizar facilidades que causem transtornos à população local como indústrias químicas, reatores nucleares, usinas etc. Propõem um método baseado em relaxação lagrangeana, que trabalha com os subproblemas de localizar e rotear.

Um método híbrido para a resolução de um PLR capacitado de dois níveis é apresentado por Prins et al. em 2006 e 2007. No primeiro trabalho, os autores resolvem heurísticamente o PLR utilizando uma meta-heurística GRASP. No segundo trabalho, a heurística utilizada para a resolução do PLR é baseada em relaxação lagrangeana e busca tabu, procurando trocar informação entre a localização e o roteamento. Os testes são realizados no banco de teste proposto pelo autor em Prins et al. (2004), no banco de teste proposto por Tuzun (1999) e a coletânea de instâncias conhecidas como “BARRETO” (Barreto, 2003). É importante ressaltar que o banco de teste proposto é apelidado de “PRINS” e é utilizado pelos trabalhos mais atuais. O banco de teste possui 30 instâncias com 20 a 200 clientes e de 5 a 10 possíveis depósitos.

Uma formulação baseada em *set-partitioning* é proposta por Akca (2009) para um PLR de dois níveis capacitado. O método é comparado com formulações baseadas em grafos utilizadas em trabalhos publicados anteriormente. Um algoritmo de *branch-and-price* é proposto e testado em algumas instâncias do banco de teste “BARRETO” e instâncias com 10 a 20 clientes.

Contardo et al. (2010) e Belenguer et al. (2010, 2011) propõem o algoritmo *branch-and-cut* para o PLR de dois níveis capacitado. Os testes são realizados nos

bancos de testes propostos por Akca et al. (2009), Belenguer et al. (2010) e “BARRETO”.

A formulação exata de maior sucesso para o PLRC de dois níveis na atualidade é o proposto por Baldacci et al. (2011). O método é baseado em *set-partitioning* e obtém os melhores resultados para os bancos de testes propostos por “BARRETO”, “AKCA”, “PRINS” e “TUZUN” com até 100 clientes. Várias soluções ótimas foram confirmadas neste trabalho.

Na Tabela 1, pode-se observar as principais publicações na área do PLR em ordem cronológica.

Tabela 1: Tabela Cronológica de Artigos

Ano	Autor	Título
1963	Maranzana, F. E.	<i>On the Location of Supply Points to Minimize Transportation Costs</i>
1964	Maranzana, F. E.	<i>On the Location of Supply Points to Minimize Transportation Costs</i>
1968	Griffiths, D. O.	<i>The Use of Regression in a Depot Location Exercise</i>
1968	Webb, M.	<i>Cost Functions in the Locations of Depots for Multiple-Delivery Journeys</i>
1969	Christofides, N. Eilon, S.	<i>Expected Distances in Distribution Problems</i>
1969	Lawrence, R. M., Pengilly, P.	<i>The Number and Location of Depots Required for Handling Products for Distribution to Retail Stores in South-East England</i>
1972	Wren, A., Holliday, A.	<i>Computer Scheduling of Vehicles from One or More Depots to a Number of Delivery Points</i>
1973	Watson-Gandy, C. D. T., Dohrn, P. J.	<i>Depot Locations with Van Salesman – A Practical Approach</i>
1976	Chan., A. W., Francis, R. L.	<i>A Round-Trip Location Problem on a Tree Graph</i>
1976	Burness, R. C., White, J. A.,	<i>The Traveling Salesman Location Problem</i>
1977	Chan., A. W., Hearn, D. W.	<i>A Rectilinear Distance Round Trip Location Problem</i>
1979	Harrison, H.	<i>A Planning System for Facilities and Resources in Distribution Networks</i>
1979	Or, I., Pierskall, W. P.	<i>A Transportation Location-Allocation Model for Regional Blood Banking</i>
1980	Jacobsen, S. K., Madsen, O. B. G.,	<i>A Comparative Study of Heuristics for a Two-Level Routing-Location Problem</i>
1981	Ghosh, J. K., Sinha, S. B., Acharya, D.	<i>A Generalized Reduced Gradient Based Approach to Round-Trip Location Problem</i>
1981	Madsen, O. B. J.	<i>A Survey of Methods for Solving Combined Location-Routing Problems</i>

1981	Laporte, G., Nobert, Y.	<i>An Exact Algorithm for Minimizing Routing and Operating Costs in Depot Location</i>
1981	Nambiar, J. M., Gelders, L. F., Wassenhove, L. N.	<i>A Large Scale Location-Allocation Problem in the Natural Rubber Industry</i>
1982	Drezner, Z., Wesolowsky G. O.	<i>A Trajectory Approach to the Round-Trip Location Problem</i>
1982	Drezner, Z.,	<i>Fast Algorithms for The Round Trip Location Problem</i>
1983	Laporte, G., Nobert, Y., Pelletier, P.	<i>Hamilton Location Problems</i>
1983	Madsen, O. B. G.	<i>Methods for Solving Combined Two Level Location Routing Problems of Realistic Dimensions</i>
1984	Perl, J., Daskin, M. S.	<i>A Unified Warehouse Location-Routing Methodology</i>
1985	Ichimori, T., Nishida, T.	<i>Note on a Rectilinear Distance Round-Trip Location Problem</i>
1985	Drezner Z.,	<i>$O(N \log N)$ Algorithm for the Rectilinear Round-Trip Location Problem</i>
1985	Kolen, A.,	<i>The Round Trip p-Center and Covering Problem on a Tree</i>
1985	Drezner, Z., Steiner, G., Wesolowsky, G. O.	<i>One-Facility Location with Rectilinear Tour Distance</i>
1985	Perl, J., Daskin, M. S.	<i>A Warehouse Location-Routing Problem</i>
1986	Laporte, G., Nobert, Y., Arpin, D.	<i>An Exact Algorithm for Solving a Capacitated Location-Routing Problem</i>
1986	Labbé, M., Laporte, G.,	<i>Maximizing User Convenience and Postal Service Efficiency in Post Box Location</i>
1986	Berman, O., Simchi-Levi, D.	<i>Minisum Location of Travelling Salesman</i>
1987	Balakrishnan, A., Ward, J. E., Wong, R. T.	<i>Integrated Facility Location and Vehicle Routing Models: Recent Work and Future Prospects</i>
1987	Daskin, M. S.	<i>Location, Dispatching and Routing Models for Emergency Services with Stochastic Travel Times</i>
1988	Laporte, G.	<i>Location-Routing Problems</i>
1988	Laporte, G., Norbert, Y., Taillefer, S.,	<i>Solving a Family of Multi-Depot Vehicle Routing and Location-Routing Problems</i>
1988	Berman, O., Simchi-Levi, D.,	<i>Finding the Optimal a Priori Tour and Location of a Traveling Salesman with Nonhomogeneous Customers</i>
1988	Simchi-Levi, D., Berman, O.	<i>A heuristic Algorithm for the Traveling Salesman Location Problem on Networks</i>
1988	Bookbinder, J., Reece, K. E.	<i>Vehicle Routing Considerations in Distribution System Design</i>
1989	Laporte, G.	<i>A Survey of Algorithms for Location-Routing Problems</i>
1989	Salhi, S., Rand, G. G.	<i>The Effect of Ignoring Routes when Locating Depots</i>

1989	Eiselt, H. A., Laporte, G.	<i>Integrated Planning in Distribution System</i>
1989	Laporte, G., Louveaux, F., Mercure, H.	<i>Models and Exact Solutions for a Class of Stochastic Location-Routing Problems</i>
1989	Levy, L., Bodin, L.	<i>The Arc Oriented Location Routing Problem</i>
1989	Laporte, G., Dejax, P. J.	<i>Dynamic Location-Routing Problems</i>
1989	Nambiar, J. M., Gelders, L. F., Wassenhove, L. N.	<i>Plant Location and Vehicle Routing in the Malaysian Rubber Smallholder Sector: A Case Study</i>
1989	Zografos, K.G., Samara, S.	<i>Combined Location-Routing Model for Hazardous Waste Transportation and Disposal</i>
1990	Berstimas, J., Jaillet, P., Odoni, A. R.	<i>A Priori Optimization</i>
1990	Srivastava, R., Benton, W. C.	<i>The Location-Routing Problem: Considerations in Physical Distribution System Design</i>
1991	ReVelle, C., Cohon, J., Shobrys, D.	<i>Simultaneous Siting and Routing in the Disposal of Hazardous Wastes</i>
1991	Simchi-Levi, D.	<i>The Capacitated Traveling Salesman Location Problem</i>
1991	List, G. F., Mirchandani, P.	<i>An Integrated Network/Planar Multiobjective Model for Routing and Siting for Hazardous Materials and Wastes</i>
1991	List, G. F., Mirchandani, P. B., Turnquist, M. A., Zografos, K. G.	<i>Modeling and Analysis for Hazardous Materials Transportation: Risk Analysis, Routing/Scheduling and Facility Location</i>
1993	Wong, D. W. S., Meyer, J. W.	<i>A Spatial Decision Support System Approach to Evaluate the Efficiency of Meals-On Wheels Program</i>
1993	Stowers, C. L., Palekar, U. S.	<i>Location Models with Routing Considerations for a Single Obnoxious Facility</i>
1993	Srivastava, R.	<i>Alternate Solution Procedures for the Location- Routing Problem</i>
1993	Chien, T. W.	<i>Heuristic Procedures for Practical-Sized Uncapacitated Location-Capacitated Routing Problems</i>
1993	Boffey, B., Karkazis, J.	<i>Models and Methods for Location and Routing Decision Relating to Hazardous Materials</i>
1994	Jamil, M., Batta, R., Malon, D.	<i>The Traveling Repairperson Home Base Location Problem</i>
1994	Averbakh, I., Berman, O.	<i>Routing and Location-Routing p-Delivery Men Problems on a Path</i>
1994	Hansen, P. H., Hegedahl, B., Hjortkjaer, S., Obel, B.	<i>A Heuristic Solution to the Warehouse Location- Routing Problem</i>
1995	Klose, A.	<i>Using Clustering Methods in Problems of Combined Location and Routing</i>
1995	Berman, O., Jaillet, P., Simchi-Levi, D.	<i>Location-Routing Problems with Uncertainty</i>
1995	Averbakh, I., Berman, O.	<i>Probabilistic Sales-Delivery Man and Sales-Delivery Facility Location Problems</i>

1996	Salhi, S., Fraser, M.	<i>An Integrated Heuristic Approach for the Combined Location Vehicle Fleet Mix Problem</i>
1996	Nagy, G., Salhi, S.	<i>Nested Heuristic Methods for the Location-Routing Problem</i>
1996	Nagy, G., Salhi, S.	<i>A Nested Location-Routing Heuristic Using Route Length Estimation</i>
1996	Min, H.	<i>Consolidation Terminal Location-Allocation and Consolidated Routing Problems</i>
1996	Derks, J. J. M., Spijksman, F. C. R.	<i>A Decision Support System for Locating Facilities and Routing Traffic on a Factory Site</i>
1997	Bruns, A., Klose, A.	<i>A "Locate First – Route Second" Heuristic for a combined Location-Routing Problem</i>
1997	Coutinho-Rodrigues, J., Clímaco, J., Current, J., Ratick, S.	<i>An Interactive Spatial Decision System for Multiobjective HAZMAT Location-Routing Problems</i>
1998	Min, H., Jayaraman, V., Srivastava, R.	<i>Combined Location-Routing Problems: A Synthesis and Future Research Directions</i>
1998	Nagy, G., Salhi, S.	<i>The Many-to-Many Location-Routing Problem</i>
1998	Giannikos, I.	<i>A Multiobjective Programming Model for Locating Treatment Sites and Routing Hazardous Wastes</i>
1999	Ghiani, G., Larporte, G.	<i>Eulerian Location Problems</i>
1999	Salhi, S., Naggy, G.	<i>Consistency and Robustness in Location-Routing</i>
1999	Tuzun, D., Burke, L. I.	<i>A Two-Phase Tabu Search Approach to the Location Routing Problem</i>
1999	Murty, K. G., Djang, P. A.	<i>The U.S. Army National Guard's Mobile Training Simulators Locations and Routing Problem</i>
2000	Johnson, M. P., Gorr, W. L., Roehrig, S. F., Heinz, J. H.	<i>Location/Allocation/Routing for Home-Delivered Meals Provision: Models and Solution Approaches</i>
2001	Chan, Y., Carter, W. B., Burness, M. D.	<i>A Multi-Depot, Multi-Vehicle, Location-Routing Problem with Stochastically Processed Demands</i>
2001	Ghiani, G., Laporte, G.	<i>Location-Arc Routing Problems</i>
2002	Wu, T., Low, C., Bai, J.	<i>Heuristic Solutions to Multi-Depot Location Routing Problems</i>
2003	Lee, Y., Kim, S., Lee, S., Kang, K.	<i>A Location-Routing Problem in Designing Optical Internet Access with WDM Systems</i>
2004	Cappanera, P., Gallo, G., Maffioli, F.	<i>Discrete Facility Location and Routing of Obnoxious Activities</i>
2004	Prins, C., Prodhon, C., Wolfier Calvo, R.	<i>Nouveaux algorithmes pour le problème de localisation et routage sous contraintes de capacité</i>
2006	Prins, C., Prodhon, C., Wolfier Calvo, R.	<i>Solving the Capacitated Location-Routing Problem by a Grasp Complemented by a Learning Process and a Path Relinking</i>
2007	Prins, C., Prodhon, C., Ruiz, A., Siriano, P., Calvo, R. W.	<i>Solving the Capacitated Location-Routing Problem by a Cooperative Lagrangean Relaxation-Granular Tabu Search Heuristic</i>
2007	Barreto S. S., Paixão, J., Santos, S. S.	<i>Using Clustering Analysis in a Capacitated Location-Routing Problem</i>

2008	Duhamel C., Lacomme, P., Prins, C., Prodhon, C.	<i>A Memetic Approach for the Capacited Location Routing Problem</i>
2010	Contardo, C., Cordeau, J., Gendron, B.	<i>A Branch-and-Cut Algorithm for the Capacitated Location-Routing Problem</i>
2010	Belenguer, J. M., Benavent, E., Prins, C., Prodhon, C., Wolfier Calvo, R.	<i>A Branch-and-Cut Algorithm for the Location-Routing Problem with Depot and Vehicle Capacities</i>
2011	Belenguer, J. M., Benavent, E., Prins, C., Prodhon, C., Wolfier Calvo, R.	<i>A Branch-and-Cut Method for the Capacited Location-Routing Problem</i>
2011	Baldacci, R., Mingozzi, A., Wolfier Calvo, R.	<i>An Exact Method for the Capacitated Location-Routing Problem</i>
2011	Hemmelmayr, V., C., Cordeau, J., Crainic, T., G.	<i>An Adaptive Large Neighborhood Search Heuristic for Two-Echelon Vehicle Routing Problems Arising in City Logistics</i>

3.3 Estudos de Caso

Nesta categoria de trabalho os autores se preocuparam, em algum momento, em aplicar o conhecimento em um caso real. No começo da Tabela 2, encontra-se o trabalho já comentado de Maranzana (1963), que trata de um problema no qual 40 cidades italianas são consideradas como clientes e utiliza três depósitos. Em seguida, encontramos o trabalho de Lawrence e Pengilly (1969) que estudam um PLR de três níveis relacionados com uma companhia localizada a sudeste da Inglaterra. O primeiro nível é composto por três fábricas, o segundo por centros de distribuição e os clientes encontram-se no terceiro nível. O objetivo do problema é localizar depósitos, determinar as suas dimensões e designar as rotas de distribuição. Por causa da dificuldade para rotear, utilizaram agrupamentos de maneira que os clientes poderiam ser atendidos por apenas um veículo. O problema de localização foi resolvido através de um procedimento iterativo.

O trabalho de Wren e Holliday (1972) trata de um problema onde uma companhia deseja decidir entre duas localizações para um novo depósito. O algoritmo proposto foi aplicado a cada uma das localizações disponíveis para o depósito e avaliados comparativamente. Os resultados permitem antever a frota necessária em cada depósito e as rotas para atendimento aos clientes. Além disso, o algoritmo produzido foi utilizado pela empresa para resolver problemas de 320 clientes e 4 depósitos.

Um trabalho real de PLR com dois níveis é apresentado pelos autores Warson-Gandy e Dohrn (1973). O problema é extraído de uma companhia do ramo alimentar, onde são avaliadas várias hipóteses de localização de depósitos que, por

tentativa e erro, procuram tomar a melhor decisão. Neste problema, a função objetivo é formada com base no resultados das vendas, custo de distribuição, custo de transporte das fábricas para os depósitos e os custos dos depósitos.

No ramo da indústria farmacêutica, Harrison (1979) estuda a distribuição de produtos de uma grande empresa na Irlanda. A abordagem sobre a localização e a distribuição é feita separadamente. Para solucionar o problema, é proposto um modelo de programação estocástica de três fases. A primeira é um gerador matricial, a segunda integra o modelo de programação matemática e a terceira realiza o roteamento. Foram testados problemas com 1200 clientes e foram atingidas economias de 23,3% nos custos de distribuição e de 20% em transporte.

Na área de saúde, Or e Pierskall (1979) estudam o problema de localização de bancos de sangue na região metropolitana de Chicago, para em seguida rotear para os hospitais. O modelo considera dois tipos de rotas: caixeiro viajante e o de ida e volta. Considera também custos relacionados com o número de requisições aos hospitais atribuídos a um certo banco de sangue. As necessidades dos hospitais são determinísticas. O problema possui número limite de veículos, considera a capacidade e um limite para o tamanho da rota. Para simplificar o modelo e facilitar sua resolução, eles consideram duas premissas: definem previamente um determinado número de bancos de sangue a instalar; e estabelecem uma periodicidade diária para entregas. Utilizam uma formulação de programação linear inteira para o PLR e uma resolução heurística iterativa de rotear e localizar.

Os autores Jacobsen e Madsen (1980) lidam com o trabalho de distribuir jornais na Dinamarca. Os autores utilizam o termo *Location-Routing Problem* pela primeira vez e, neste caso, o PLR possui três níveis. A partir do local de impressão, os jornais são enviados através de veículos para pontos de distribuição que podem variar de um a três para em seguida serem redistribuídos para os 4510 clientes. Os veículos utilizados para o traslado entre a fábrica e os centro de distribuição possuem maior capacidade que os utilizados para distribuir aos clientes. Os pontos de distribuição podem ser abertos sem nenhum custo. O custo das rotas é calculado por um custo fixo e um custo dependente da distância. Os clientes possuem janela de tempo com apenas tempo limite. Também são introduzidas taxas de produção, capacidade e duração. Para resolver este problema, os autores não utilizaram artifícios de simplificação. Utilizaram três heurísticas: ALA-SAV, SAV-DROP e TREE-TOUR. A heurística ALA-SAV é uma combinação do método ALA de

localização e atribuição com o método SAV, que é um escalonamento de veículos do tipo *saving* a cada depósito e seus clientes. A heurística SAV-DROP agrupa os clientes nas rotas utilizando o método *saving*, e o procedimento para testar outros depósitos para as rotas foi chamado de DROP. A heurística TREE-TOUR constrói uma árvore de rotas possuindo todos os clientes e as localizações dos depósitos. Esta última heurística obteve os piores resultados. Em 1983, Madsen publica um trabalho complementar com uma revisão sobre a literatura, uma revisão das abordagens heurísticas e uma proposta de solução partindo do problema matemático para solucionar o problema de distribuição de jornais tratado no artigo citado anteriormente.

O trabalho de Nambiar et al. (1981) estuda a localização de fábricas de processamento de borracha natural da Malásia. Neste caso, as estações de coleta são os clientes a serem roteados. O objetivo do problema é minimizar os custos de roteamento, as horas extras, os custos de operação das fábricas e dos veículos. O problema possui as restrições de limite de frota por fábrica, capacidade dos veículos e fábrica, frota fixa por estação de coleta, tempo de viagem entre os clientes, tempo máximo por rota, tempo máximo de horas extras por dia e janelas de tempo devido à característica perecível do produto. Para a resolução, utiliza-se uma heurística do tipo localiza e roteia baseada no método *saving*. Outro algoritmo foi construído para resolver o problema quando as estações de coleta forem esparsas. Nesta versão, o algoritmo não agrupa as estações por área e procura tirar vantagem do roteamento. A instância construída com dados reais possui 50 estações de coleta e seis localizações de fábricas. Em 1989 o problema volta a ser estudado pelos autores. O estudo do problema é aprofundado e as etapas de localização, atribuição e roteamento das rotas são separadas. Neste trabalho, passa-se a dar um foco maior ao roteamento dos clientes, tornando o planejamento diário. O problema real estudado contém 50 estações de coleta que servem a uma fábrica. As distâncias percorridas são reduzidas em até 32%. Para resolver uma instância com 89 estações de coleta e 10 possíveis localizações de fábricas, aplicam um procedimento dual para resolver o problema de localização e atribuição, relaxando as restrições de capacidade das fábricas e de tempo das rotas. Os autores chegam à conclusão de que se abrissem mais duas fábricas, totalizando três, as economias nos custos poderiam chegar a 50%.

Um estudo de caso em cima da distribuição dos correios foi publicado por Levy e Bodin (1989). O PLR foi modelado orientado para arcos considerando que as ruas são os clientes. Os depósitos são considerados os veículos dos carteiros de onde se iniciará o percurso a pé. Para solucionar o problema, é utilizado uma heurística do tipo localiza, atribui e roteia. O problema foi testado em uma instância com 206 nós e 564 arcos.

Na categoria de resíduos, Re Velle (1991) apresenta um PLR que trata de resíduos nucleares. Diferentemente da maioria dos trabalhos, a função objetivo deste problema é reduzir o perigo à população e à carga transportada. O estudo foi realizado para o Departamento de Energia dos Estados Unidos e teve como instância 14 centrais nucleares e 24 potenciais centros de coleta situados na zona leste dos Estados Unidos.

Ainda em resíduos, List e Mirchandani (1991) tratam do problema de transporte e armazenagem de lixos e materiais perigosos no distrito de Albania. Neste problema consideram o risco e o custo. Utilizam um sistema de suporte a decisão para resolver o problema e testam em dados da *Capital District*. A instância possui 124 arcos, 86 nós e 15 produtores de lixo. Existem quatro pontos para a abertura de depósitos e uma capacidade limite. As simplificações consideradas são apenas um tipo de lixo, um único grau de perigo, estações de tratamento do mesmo tipo e reduzem os pontos de impacto a um.

O trabalho de Wong e Meyer (1993) trata de um problema de distribuição de alimentos para idosos carentes na região de Connecticut. Os autores utilizam um pacote comercial chamado *TransCad* para resolver o problema. O objetivo é determinar rotas ótimas para os distribuidores de alimentos, submetido às restrições de tempo e capacidade, além de determinar a localização para a cozinha. Eles usaram a abordagem dividir para conquistar resolvendo em separado vários componentes do problema.

Derks e Spieksman (1996) realizam um estudo de um PLR relacionado com uma empresa de cimento da Holanda. O objetivo é determinar a localização de serviços no interior da empresa e as rotas de fluxo, minimizando o congestionamento. A rede considerada é orientada, com restrições de capacidade nos vértices e nos arcos, além de caminhos mandatórios. Para solucionar o problema, uma heurística de busca local é utilizada.

Uma aplicação militar é estudada no trabalho de Murty e Djang (1999). O problema consiste em decidir a localização de 21 instalações para simuladores de treinos entre 29 possíveis locais. Esses simuladores se deslocarão para localizações secundárias onde proverão os treinamentos. O problema é dividido em vários problemas menores e é resolvido heurísticamente. Como resultado, as distâncias percorridas foram 70% menores que o planejamento inicial.

Outro trabalho em cima do problema de distribuição de alimentos para idosos carentes é publicado por Johnson et al. (2000). O trabalho de distribuição é voluntário em Allegheny County no subúrbio de Pittsburgh. O objetivo do trabalho é localizar as cozinhas e as rotas de distribuição. A heurística para solucionar o problema realiza localização e atribuição para depois rotear.

A Tabela 2 apresenta os artigos apresentados nesta seção em ordem cronológica.

Tabela 2: Estudos de Caso - Ordem Cronológica

Ano	Autor	Título
1963	Maranzana, F. E.	<i>On the Location of Supply Points to Minimize Transportation Costs</i>
1969	Lawrence, R. M., Pengilly, P.	<i>The Number and Location of Depots Required for Handling Products for Distribution to Retail Stores in South-East England</i>
1972	Wren, A., Holliday, A.	<i>Computer Scheduling of Vehicles from One or More Depots to a Number of Delivery Points</i>
1973	Watson-Gandy, C. D. T., Dohrn, P. J.	<i>Depot Locations with Van Salesman – A Practical Approach</i>
1979	Harrison, H.	<i>A Planning System for Facilities and Resources in Distribution Networks</i>
1979	Or, I., Pierskall, W. P.	<i>A Transportation Location-Allocation Model for Regional Blood Banking</i>
1980	Jacobsen, S. K., Madsen, O. B. G.,	<i>A Comparative Study of Heuristics for a Two-Level Routing-Location Problem</i>
1981	Nambiar, J. M., Gelders, L. F., Wassenhove, L. N.	<i>A Large Scale Location-Allocation Problem in the Natural Rubber Industry</i>
1983	Madsen, O. B. G.	<i>Methods for Solving Combined Two Level Location Routing Problems of Realistic Dimensions</i>
1989	Nambiar, J. M., Gelders, L. F., Wassenhove, L. N.	<i>Plant Location and Vehicle Routing in the Malaysian Rubber Smallholder Sector: A Case Study</i>
1989	Levy, L., Bodin, L.	<i>The Arc Oriented Location Routing Problem</i>
1991	Re Velle, C., Cohon, J., Shobrys, D.	<i>Simultaneous Siting and Routing in the Disposal of Hazardous Wastes</i>
1991	List, G. F., Mirchandani, P.	<i>An Integrated Network/Planar Multiobjective Model for Routing and Siting for Hazardous Materials and Wastes</i>

1993	Wong, D. W. S., Meyer, J. W.	<i>A Spatial Decision Support System Approach to Evaluate the Efficiency of Meals-On Wheels Program</i>
1996	Derks, J. J. M., Spijksman, F. C. R.	<i>A Decision Support System for Locating Facilities and Routing Traffic on a Factory Site</i>
1999	Murty, K. G., Djang, P. A.	<i>The U.S. Army National Guard's Mobile Training Simulators Locations and Routing Problem</i>
2000	Johnson, M. P., Gorr, W. L., Roehrig, S. F., Heinz, J. H.	<i>Location/Allocation/Routing for Home-Delivered Meals Provision: Models and Solution Approaches</i>
2003	Lee, Y., Kim, S., Lee, S., Kang, K.	<i>A Location-Routing Problem in Designing Optical Internet Access with WDM Systems</i>

3.4 Estudos de Revisão

Os trabalhos apresentados nesta seção se preocupam com revisão bibliográfica, taxonomias, classificação e avaliação de métodos de resolução. Na Tabela 3 os trabalhos são apresentados em ordem cronológica.

Tabela 3: Estudos de Revisão – Ordem Cronológica

Ano	Autor	Título
1981	Madsen, O. B. J.	<i>A Survey of Methods for Solving Combined Location-Routing Problems</i>
1983	Madsen, O. B. G.	<i>Methods for Solving Combined Two Level Location Routing Problems of Realistic Dimensions</i>
1987	Balakrishnan, A., Ward, J. E., Wong, R. T.	<i>Integrated Facility Location and Vehicle Routing Models: Recent Work and Future Prospects</i>
1988	Laporte, G.	<i>Location-Routing Problems</i>
1989	Laporte, G.	<i>A Survey of Algorithms for Location-Routing Problems</i>
1989	Eiselt, H. A., Laporte, G.	<i>Integrated Planning in Distribution System</i>
1998	Min, H., Jayaraman, V., Srivastava, R.	<i>Combined Location-Routing Problems: A Synthesis and Future Research Directions</i>
2007	Nagy, G. e Salhi, S.	<i>Location-routing: Issues, models and methods</i>

3.5 Resumo

O PLR é estudado há quase 50 anos com mais de 100 publicações em diversas mídias, podendo ser dividido em estudo teórico e estudos de caso. Possui um maior número de propostas heurísticas para resolução do problema quando comparado com os métodos exatos propostos devido à capacidade das heurísticas de prover uma solução para instâncias com mais de 100 nós. Os métodos propostos

com os melhores resultados utilizam troca de informações entre o roteamento e a localização. Algoritmos que utilizam a abordagem que particiona o problema em duas tomadas de decisão, localização e roteamento, logo foram superados. Outro ponto a ser considerado nas heurísticas propostas é a falta do fator aleatório. Os algoritmos utilizam este fator apenas para gerar uma solução inicial e a maioria das tomadas de decisões posteriores leva em conta, na maioria das vezes, o fator localização, levando a ótimos locais. Utilizar aleatoriedade nessas fases economiza processamento devido a não ter cálculos para decidir que próximo passo realizar. O capítulo a seguir mostrará que as perturbações de maior impacto do algoritmo proposto sorteiam aleatoriamente os depósitos a serem trocados, além das perturbações já conhecidas na literatura entre clientes. Versões paralelas de algoritmos para o PLR não foram encontradas na literatura.

Capítulo 4 – Algoritmo Proposto

O algoritmo proposto é baseado na meta-heurística *Iterated Local Search* (ILS), que basicamente inicia com uma heurística de construção seguida de uma busca local e perturbações. Para encontrar outros ótimos locais, após a etapa da busca local é efetuada uma perturbação sobre a solução incumbente e esta solução perturbada passa por um critério de aceitação antes de retornar ao procedimento de busca local. Segundo Lourenço et al. (2002), a ideia essencial da ILS está atribuída ao fato desta meta-heurística realçar seu foco em um subespaço menor, ao invés de considerar o espaço completo de soluções, definido por aquelas que são ótimos locais de um dado procedimento de otimização. Uma ilustração gráfica representando o comportamento de uma solução “s” no procedimento ILS pode ser observada na Figura 4.

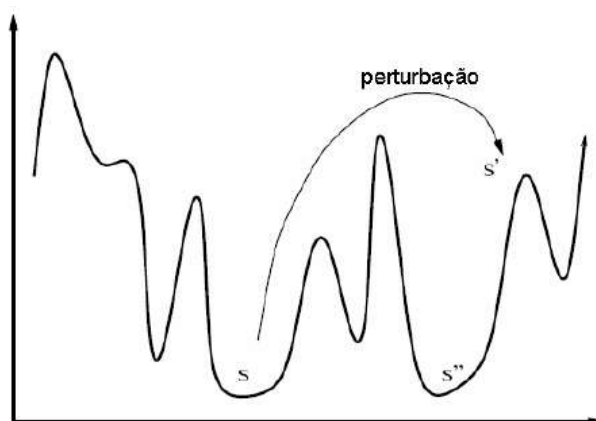


Figura 4: Gráfico do comportamento da solução no procedimento ILS (Lourenço, 2002).

Para se aplicar a meta-heurística ILS, quatro procedimentos devem ser especificados, conforme pode ser observado no Algoritmo 1. São eles:

1. “Construcao”, no qual uma solução inicial é construída;
2. “BuscaLocal”, realiza buscas locais na solução inicialmente obtida;
3. “Perturbacao”, onde um novo ponto de partida é gerado, por meio de uma perturbação de uma solução encontrada na busca local;

4. “CritérioAceitacao”, que determina a partir de qual solução deve-se prosseguir a busca.

Algoritmo 1. ILS

1. $s \leftarrow \text{Construcao};$
 2. $s \leftarrow \text{BuscaLocal}(s_0);$
 3. **Enquanto** (critério de parada não obedecido) **faça**
 4. $s' \leftarrow \text{Pertubacao}(s, \text{histórico});$
 5. $s'' \leftarrow \text{BuscaLocal}(s');$
 6. $s \leftarrow \text{CritérioAceitacao}(s, s'', \text{histórico})$
 7. **Retorna**(s)
-

De acordo com Stützle (1998), a modificação efetuada na etapa de perturbação é empregada com o objetivo de escapar de uma solução ótima local ainda distante de um ótimo global. Frequentemente, o movimento é escolhido aleatoriamente dentro de uma vizinhança de maior entropia em relação à utilizada no procedimento de busca local, ou um movimento que o algoritmo de busca local não possa reverter em apenas um passo. A princípio, qualquer método de busca local pode ser utilizado, porém seu desempenho, no tocante à qualidade da solução e esforço computacional, depende fortemente do algoritmo escolhido. Geralmente uma heurística de descida (em problemas de minimização) é usada, contudo a adoção de um procedimento mais complexo também é admissível.

O critério de aceitação é utilizado para decidir qual a próxima solução a ser perturbada. A escolha deste critério é importante, pois controla o balanço entre a intensificação e diversificação. O histórico de busca é usado para decidir se alguma solução ótima local, encontrada anteriormente, deve ser escolhida ou se exerce algum tipo de influência na decisão entre s e s'' .

O procedimento ILS deve conduzir a boas amostragens do espaço de busca desde que as perturbações não sejam nem tão grandes nem demasiadamente pequenas. Se forem pequenas, poucas novas soluções serão exploradas, ao passo que se forem muito grandes, o algoritmo tenderá adotar pontos de partidas praticamente aleatórios.

Atualmente a literatura nos tem mostrado que as melhores versões das heurísticas ILS são aquelas em que a busca local é efetuada por uma estrutura do tipo VNS ou VND (Penna et.al. 2012, Mine et.al. 2011). Neste contexto, no ILS aqui proposto, a fase de busca local é efetuada por um algoritmo do tipo VND (Método de

Descida em Vizinhaça Variável – *Variable Neighborhood Descent*) proposto por Mladenović e Hansen (1997). Tal método é essencialmente caracterizado por realizar mudanças sistemáticas de estruturas de vizinhaça pertencentes a um conjunto $N = \{N_1, N_2, N_3, \dots, N_j\}$, de forma determinística. A vizinhaça $k' = k + 1 \in N$ é selecionada apenas se não houver melhora nas k vizinhanças pesquisadas anteriormente. Estruturas de vizinhanças clássicas são utilizadas: *shift*, *swap* (1,1), *swap* (2,1), *swap* (2,2), e uma adaptação do movimento *cross*, que serão explicadas na seção 4.3.

O pseudocódigo do procedimento é descrito no Algoritmo 2. O algoritmo começa definindo a frota e o número de depósitos mínimos que atendam a instância. O procedimento é bastante simples. Para calcular a frota (linha 1, algoritmo 2), a demanda total é dividida pela capacidade do veículo e utiliza-se o teto do resultado da expressão 4.1. Lembrando que q_i é a demanda associada ao cliente i e Q é a capacidade dos veículos. A única diferença para calcular os depósitos (linha 2, algoritmo 2) é que se utiliza o depósito de maior capacidade para o cálculo. Esses dados servirão de entrada para a fase de construção.

$$Frota = \frac{\sum q_i}{Q} \quad (4.1)$$

Em seguida, o algoritmo entra em um laço, que irá incrementar o número de depósitos mínimos (linha 4, algoritmo 2). Dentro deste laço, encontramos outros dois laços, sendo um o laço principal do ILS (linha 5, algoritmo 2) onde se realizam as construções (linha 6, algoritmo 2) e a primeira fase de busca locais, e o outro, o laço onde fica a segunda fase de busca locais (linha 13, algoritmo 2).

Além disso, o ILS aqui proposto é do tipo *multi-start* com sementes aleatórias para decidir qual construção será realizada em cada iteração, a ordem das buscas entre vizinhanças e as perturbações. Suas perturbações utilizam movimentos *shift* e *swap* entre depósitos e entre clientes. Podem-se destacar os seguintes procedimentos, que serão explanados em maior detalhes a seguir: construção, perturbações e RVND (VND com sequenciamento aleatório de vizinhanças).

4.1 Construção

A fase de construção (linha 6, algoritmo 2) é onde se formam as soluções a serem refinadas. Foram utilizadas duas estratégias gulosas: a inserção paralela e a inserção alternada e duas estratégias baseada em *First Fit Decreasing*. Em cada iteração do algoritmo é sorteada qual das construções será utilizada.

Algoritmo 2. ILS-RVND(Instancia)

```

1.  frota ← Estima_Frota(Instancia)
2.  depositosEstimados ← Estima_Depositos(Instancia)
3.  s, s', s'' ← 0
4.  Para depositosEstimados < Instancia.Depositos ou rodada sem melhora faça
5.      Para i ← 0 até i < MAX_ITER faça
6.          s ← Construcão(frota, depositosEstimados);
7.          s ← RVND (s);
8.          Para j ← 0 até j < MAX_ILS1 faça
9.              s' ← Pertubacao1(s, histórico);
10.             s'' ← RVND(s');
11.             s ← CriterioAceitacao(s,s'', histórico)
12.             Se (s==s'') faça j ← 0
13.         Para j ← 0 até j < MAX_ILS2 faça
14.             s' ← Pertubacao2(s, histórico);
15.             s'' ← RVND(s');
16.             s ← CriterioAceitacao(s,s'', histórico)
17.             Se (s==s'') faça j ← 0
18.  Retorna(s)

```

4.1.1 Inserção Paralela

O procedimento começa com o sorteio dos depósitos de onde cada rota partirá e, após realizar todas as entregas, retornará (linha 1, algoritmo 3). Este sorteio depende de dois dados: a estimativa de depósitos e a capacidade dos veículos. Baseando-se na estimativa de depósitos, é sorteada uma quantidade dentre os depósitos disponíveis. Em seguida tais depósitos são distribuídos entre as rotas e não podem ser designados para um número maior que o teto entre a divisão da capacidade do depósito e a capacidade dos veículos. Um cliente é sorteado para

cada rota para garantir a geração de diferentes soluções iniciais (linha 2, algoritmo 3).

Definido os depósitos para cada rota, verifica-se entre cada cliente ainda não inserido nas rotas qual a melhor rota e posição para inseri-lo usando o critério custo. Definido o mais barato, insere-se o cliente na rota e o procedimento é novamente realizado para os clientes restantes. Se sobrar clientes que não puderam ser inseridos devidos a restrições de capacidade, um depósito é sorteado e uma nova rota é criada. O pseudocódigo do procedimento é apresentado no Algoritmo 3.

Algoritmo 3. Insercao_Paralela(instância, s)

```

1. Sorteia_Depositos(s)
2. Sorteia_Clientes(s)
3. Enquanto Lista_Clientes  $\neq$  0 faça
4.     Para cada cliente faça
5.         Para cada rota faça
6.             Para cada posição na rota faça
7.                 custo  $\leftarrow$  Calcula_Insercao(rota, posição, cliente)
8.                 Guarda_Melhor_Insercao(custo, rota, posição, cliente)
9.             Insere_cliente(custo, rota, posição, s, cliente)
10.    Se não consegue inserir todos os clientes faça
11.        Adiciona_Rota(s)
12. Retorna(s)

```

4.1.2 Inserção Alternada

O procedimento de construção de inserção alternada possui semelhanças com a inserção paralela. A diferença se apresenta durante a fase de inserção dos clientes nas rotas. Verifica-se para a primeira rota qual é a inserção de cliente mais barata e se insere o cliente se possível. Depois se repete o procedimento para a segunda rota e assim por diante até que todos os clientes estejam inseridos. Caso não seja possível inserir todos os clientes devido às restrições de capacidade, uma nova rota é inserida assim como está descrito na inserção paralela. O pseudocódigo do procedimento é apresentado no Algoritmo 4.

Algoritmo 4. Insercao_Alternada(Instância, s)

1. Sorteia_Depositos(s)
 2. **Se** depósitos = 1 **faça**
 3. Sorteia_Clientes(s)
 4. **Enquanto** Lista_Clientes ≠ 0 **faça**
 5. **Para** cada rota **faça**
 6. **Para** cada cliente **faça**
 7. **Para** cada posição na rota **faça**
 8. custo ← Calcula_Insercao(rota, posição, cliente)
 9. Guarda_Melhor_Insercao(custo, rota, posição, cliente)
 10. Insere_cliente(custo, rota, posição, s, cliente)
 11. **Se** não consegue inserir todos os clientes **faça**
 12. Adiciona_Rota(s)
 13. **Retorna**(s)
-

4.1.3 First Fit Decreasing

Os procedimentos de construção baseados em *First Fit Decreasing* são utilizados para construir soluções com o menor número de depósitos possíveis para instâncias com carga apertada. Carga apertada é a característica de algumas instâncias possuírem solução ótima com demanda com valores próximos a capacidade dos depósitos ou veículos. Para determinar quando a construção *First Fit Decreasing* será utilizada, um cálculo sobre a folga da carga é utilizada. Esse cálculo é feito utilizando o somatório das demandas dos clientes dividido pela multiplicação da capacidade do depósito de maior capacidade pelo teto da divisão da demanda total pela capacidade do depósito de maior capacidade (observar expressão 4.2). Esta equação retorna valores entre 0 e 1.

$$FOLGA = \frac{\sum q_i}{w_x * teto(\sum q_i \div w_x)} \quad (4.2)$$

onde w_x é a capacidade do depósito de maior capacidade.

Quando o valor retornado por este cálculo for maior ou igual a 0,85, utilizamos as construções baseadas em *First Fit Decreasing*. Foram implementadas duas versões. Na primeira, o critério de escolha dos depósitos a serem sorteados entre as rotas são determinados por suas capacidades. Quanto maior, mais chances

de utilizarmos nas rotas. Escolhidos os depósitos, tenta-se inserir os clientes de maior demanda na primeira rota, depois na segunda e assim por diante.

Na segunda versão, muito semelhante à primeira, muda-se o critério de escolha dos depósitos. Os depósitos escolhidos são os de menor custo e o critério de inserção dos clientes é a carga. Assim, o procedimento é parecido com a construção paralela. O pseudocódigo do procedimento é apresentado no Algoritmo 5.

Algoritmo 5. Insercao_FFD(instância, s)

```

1. Sorteia_Depositos(s)
2. Sorteia_Clientes(s)
3. Enquanto Lista_Clientes  $\neq$  0 faça
4.     Para cada cliente faça
5.         Para cada rota faça
6.             Para cada posição na rota faça
7.                 carga  $\leftarrow$  Calcula_Insercao(rota, posição, cliente)
8.                 Guarda_Melhor_Insercao(carga, rota, posição, cliente)
9.                 Insere_cliente(custo, rota, posição, s, cliente)
10.    Se não consegue inserir todos os clientes faça
11.        Adiciona_Rota(s)
12. Retorna(s)

```

4.2 RVND

A diferença entre o *Randomized Variable Neighborhood Search* (RVND) para o VND definido por Mladenović e Hansen (1997) e o implementado neste algoritmo é que na primeira, a escolha de vizinhança não é determinística. Utiliza-se uma semente aleatória para determinar a ordem em que as vizinhanças entre rotas serão executadas (linha 3, algoritmo 6). Caso a vizinhança sorteada seja bem sucedida, buscas intra-rota (linha 6, algoritmo 6) são realizadas nas rotas modificadas, caso contrário, a vizinhança é removida da lista da busca entre rotas. O pseudocódigo do procedimento, que é chamado no Algoritmo 2, nas linhas 7, 10 e 15, é apresentado na Algoritmo 6.

Algoritmo 6. RVND(Instância, s)

-
1. Lista_Vizinhança[];
 2. **Enquanto** Lista_Vizinhança não estiver vazia **faça**
 3. Sorteia_Vizinhança(Lista_Vizinhança)
 4. **Se** vizinhança não for bem sucedida **faça**
 5. Apague_Vizinhança(Lista_Vizinhança)
 6. **Senão** BuscaIntraRota(Instância, s)
 7. **Retorna**(s)
-

Estão presentes no algoritmo duas vizinhanças do tipo *shift*:

- *Shift 1*: consiste em remover um cliente de uma rota e inserir em outra. Na Figura 5.a pode-se observar que o cliente “7” foi inserido na outra rota entre os clientes “2” e “8”;
- *Shift 2*: consiste em remover dois clientes de uma rota e inserir em outra. Na Figura 5.b, os cliente “7” e “11” foram inseridos na outra rota entre os clientes “2” e “8”.

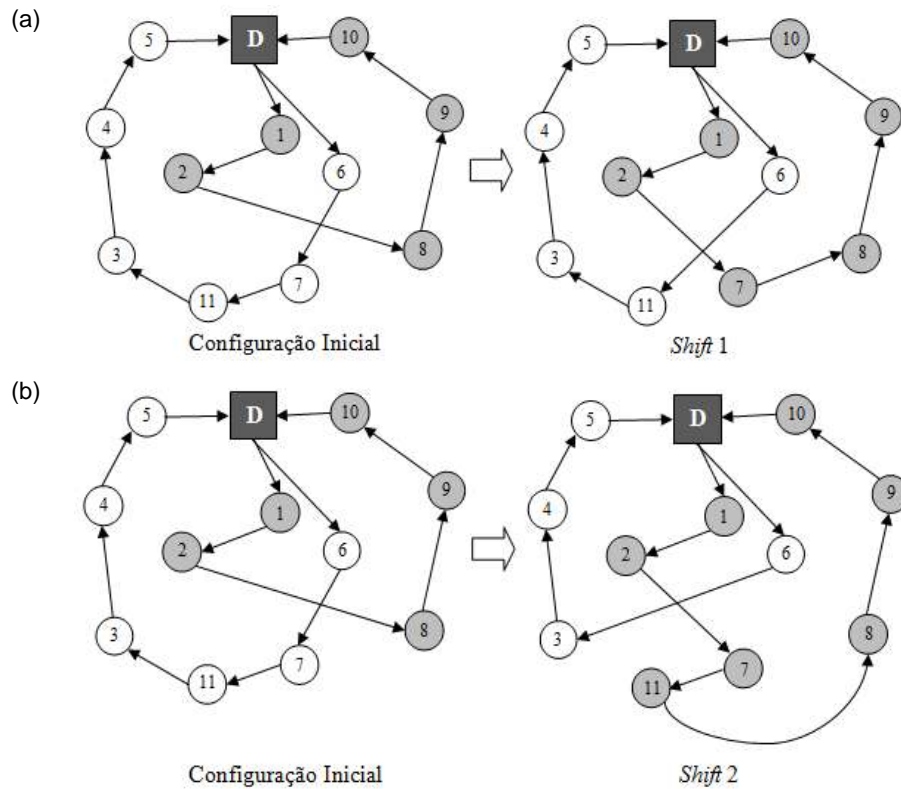


Figura 5: Vizinhança *Shift 1* e *Shift 2*

Existem três vizinhanças do tipo *swap*:

- *Swap* (1,1): consiste em trocar um cliente de uma rota com um cliente de outra. Na Figura 6.a, o cliente “2” troca de lugar com o cliente “6”;
- *Swap* (2,1): consiste em trocar dois clientes de uma rota com um cliente de outra. Na Figura 6.b os clientes “6” e “7” são trocados pelo cliente “2”;
- *Swap* (2,2): consiste em trocar dois clientes de uma rota com dois clientes de outra. Na Figura 6.c os clientes “1” e “2” são trocados pelos clientes “6” e “7”.

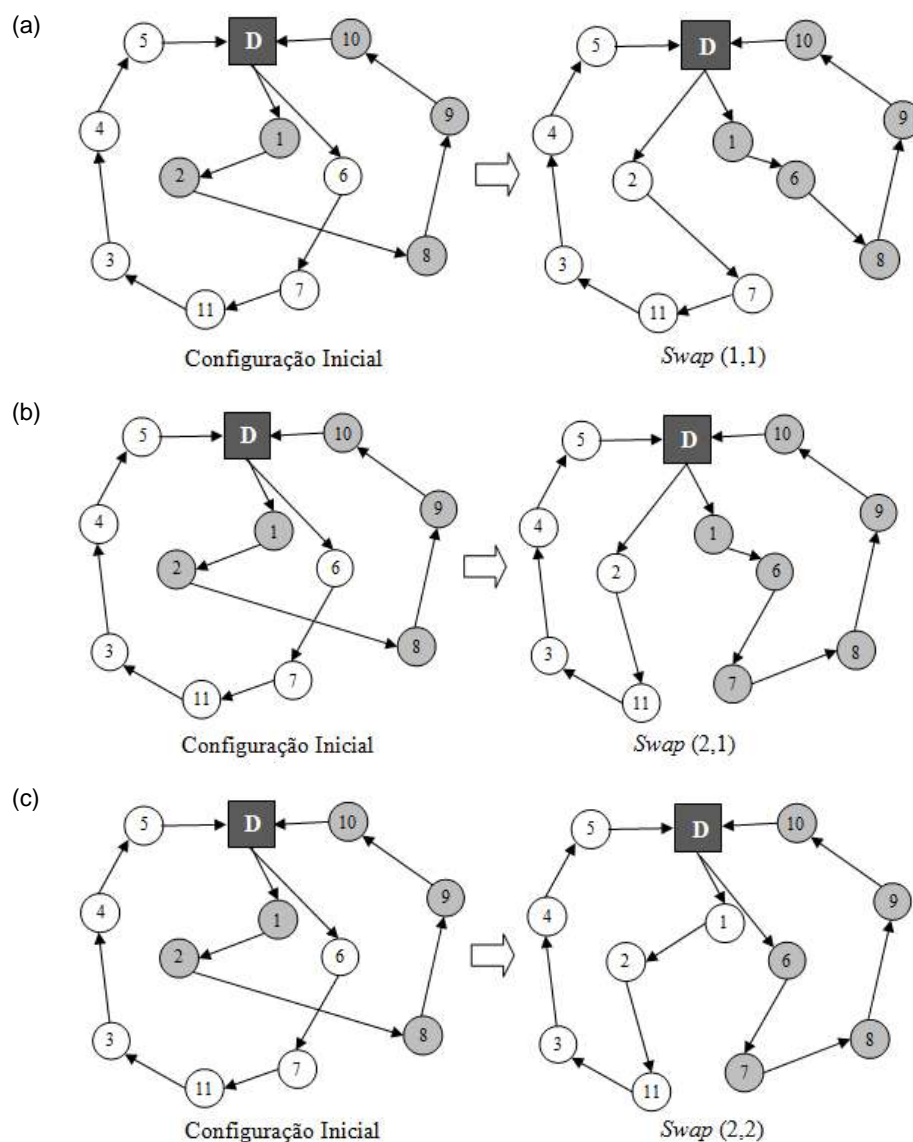


Figura 6: Vizinhança *Swap* (1,1), *Swap* (2,1) e *Swap* (2,2)

Além de um movimento de *cross*:

- *Cross*: O arco entre os clientes adjacentes c_1 e c_2 , pertencentes a uma rota r_1 , e o arco entre clientes c_3 e c_4 de uma rota r_2 são removidos. Em seguida, um arco é inserido entre c_1 e c_4 e outro entre c_3 e c_2 . Caso as rotas possuam depósitos diferentes, os arcos que retornam ao depósito são corrigidos. Na Figura 7 as arestas entre os clientes “2” e “8” e entre os clientes “7” e “11” são removidas e as arestas entre os clientes “2” e “11” e entre os clientes “7” e “8” são inseridos.

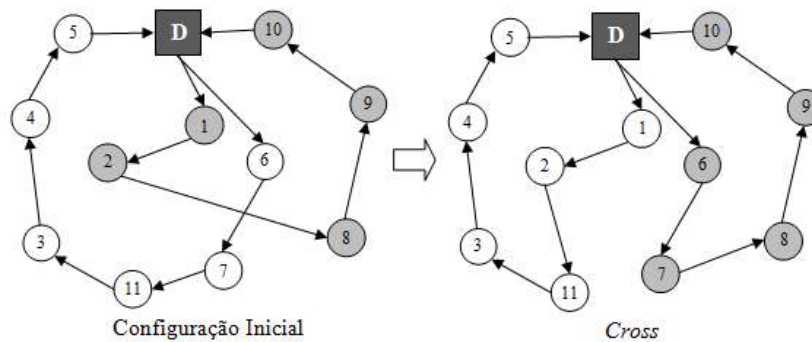


Figura 7: Vizinhança *cross*

As buscas intra-rotas (Algoritmo 7) são realizadas nas rotas que sofreram modificações pelas vizinhanças entre rotas. Assim como no método RVND, utiliza-se uma semente aleatória para determinar a ordem em que as vizinhanças intra-rotas serão executadas. Os métodos utilizados são:

- *Or-Opt 1, 2 e 3*: Proposto por Or (1976), onde um, dois ou três clientes consecutivos são removidos e posteriormente inseridos em outra posição;
- *2-optimal*: Um par de arcos é removido e outro par é inserido;
- *Permutação*: permuta dois clientes.

A complexidade das vizinhanças inter-rotas e entre rotas é $O(n^2)$.

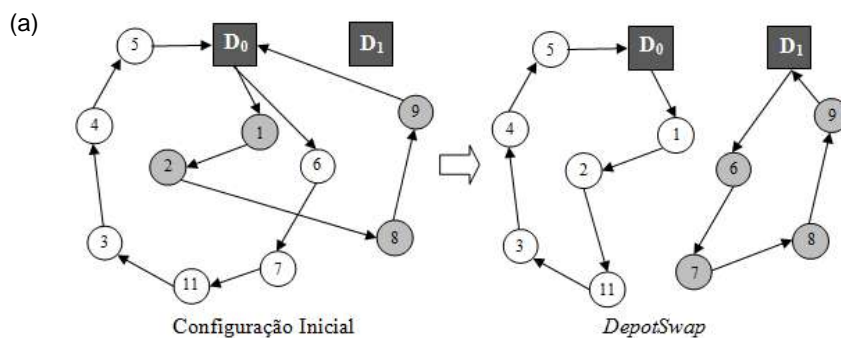
Algoritmo 7. BuscaIntraRota(Instância, s)

1. Lista_Vizinhança_IR[]
 2. **Enquanto** Lista_Vizinhança_IR não estiver vazia **faça**
 3. Sorteia_Vizinhança(Lista_Vizinhança_IR)
 4. **Se** vizinhança não for bem sucedida **faça**
 5. Apague_Vizinhança(Lista_Vizinhança_IR)
 6. **Retorna**(s)
-

4.3 Perturbações

Como já foi apresentada anteriormente, a meta-heurística ILS aplica repetidamente uma busca local às soluções iniciais obtidas através de perturbações das soluções ótimas locais previamente visitadas. No algoritmo proposto, as perturbações são movimentos de esforço computacional pequeno, todos baseados em movimentos de busca-local. Diferentemente das buscas locais, as perturbações conseguem produzir maiores alterações na solução corrente. O critério de aceitação do movimento de perturbação é que ele não pode gerar uma solução inviável. Como consequência, podem-se gerar soluções com custos piores que os das soluções já encontradas, porém permite que o algoritmo escape de mínimos locais de baixa qualidade. Foram quatro movimentos utilizados, sendo eles:

- *MultiShift*: Sorteia-se um cliente pertencente a qualquer rota e insere-o em uma outra qualquer;
- *MultiSwap*: Sorteiam-se dois clientes de rotas distintas e realiza-se a permuta;
- *DepotSwap*: Sorteia-se uma rota e troca-se o depósito; Na Figura 8.a pode-se observar a troca do depósito de uma das rotas.
- *AllDepotSwap*: Sorteiam-se as rotas que saem de um mesmo depósito e troca-se o depósito de partida por um depósito que não possua rotas. Na Figura 8.b pode-se observar que todas as rotas do depósito “D₀” são transferidas para o depósito “D₁”.



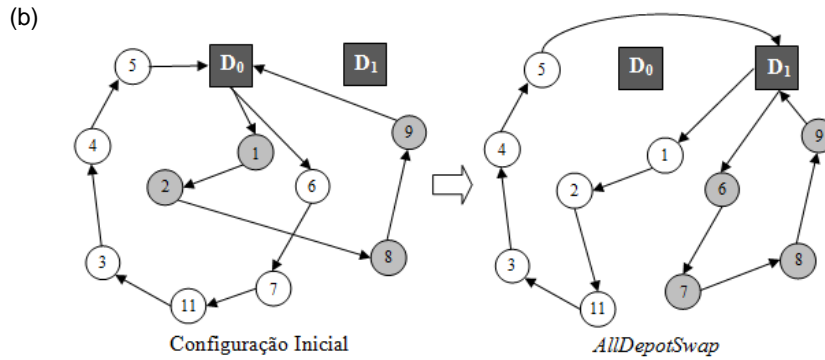


Figura 8: Perturbações entre Depósitos

As perturbações realizadas pelo procedimento Perturba1 não possuem a mesma probabilidade de ocorrer. As perturbações entre depósitos possuem 50% menos chance de ocorrer que as entre clientes (linha 9, Algoritmo 2). No procedimento Perturba2, realizam-se apenas os movimentos Multishift e MultiSwap com a mesma probabilidade de ocorrência (linha 14, Algoritmo 2).

4.4 Resumo

Neste capítulo apresentou-se o algoritmo proposto ILS-RVND, que utiliza três procedimentos de construir soluções: inserção paralela; inserção alternada; e uma construção baseada em *First Fit Decreasing* para instâncias com capacidades apertadas. A busca local do algoritmo contém seis movimentos de vizinhança entre rotas (Shift1, Shift2, Swap (1,1), Swap (2,1), Swap (2,2) e Cross) e cinco vizinhanças intra-rota (Or-opt 1, Or-opt 2, Or-opt 3, 2-optima, e Permutação). O algoritmo utiliza dois tipos de perturbações entre clientes (*Multishift1* e *Multishift2*) e dois tipos de perturbações entre depósitos (*DepotSwap* e *AllDepotSwap*). Diferentemente do ILS tradicional, uma segunda fase de refinamento foi adicionada com a característica principal de melhorar o roteamento. É um procedimento bastante parecido com a primeira fase de refinar, porém as únicas perturbações que podem ser realizadas são as entre clientes. Outra diferença importante para outros ILS-RVND publicados por outros autores é que as perturbações no primeiro refinamento tem diferentes probabilidades de ocorrer sob a justificativa de que a ocorrência de perturbações entre depósitos seguidas não levam a uma melhora na solução. Após um movimento de um depósito é necessário investigar melhor as rotas antes de realizar outro movimento entre depósitos.

No capítulo a seguir, apresentam-se os resultados obtidos em cima dos bancos de testes encontrados na literatura, apelidados por “BARRETO”, “PRINS” E “TUZUN”. São realizadas comparações com os melhores resultados da literatura e o algoritmo ALNS proposto por Hemmelmayr (2011).

Capítulo 5 – Resultados

Como citado anteriormente, os experimentos foram realizados em três bancos de testes. Um deles é composto por 19 instâncias, algumas obtidas da literatura (Or, 1976, Perl, 1983) e outras adaptadas de instâncias relacionadas com o PRV (Gaskell, 1967; Christofides e Eilon, 1969; Min, 1992; Daskin, 1995). O segundo banco de teste, proposto por Prins et al. (2004), contém 30 instâncias e o número de clientes varia de 20 a 200 e o número de depósitos de 5 a 10, além disso, possui custo para cada veículo utilizado. O terceiro banco de teste denotado por “TUZUN” pela literatura contém instâncias com depósitos com capacidade ilimitada foi proposto por Tuzun e Burke (1999). Os resultados são comparados com os resultados disponíveis na literatura (Prins, 2007; Barreto, 2007; Akca, 2008; Duhamel, 2008; Sahraeian, 2009; Contardo, 2010; Baldacci, 2011; Belenguer, 2011; Hemmelmayer, 2011). Para avaliar o algoritmo, duas comparações foram utilizadas:

- Comparam-se as melhores soluções encontradas pelo algoritmo e a média dos resultados em 10 execuções com as melhores soluções encontradas na literatura (Seção 5.1);
- Comparam-se as melhores soluções encontradas pelo algoritmo e a média dos resultados em 10 execuções com as melhores soluções da heurística ALNS proposta por Hemmelmayer et al. (2011), com a finalidade de comparar uma relação entre tempo e qualidade da solução. O algoritmo ALNS foi selecionado por possuir as melhores soluções na média para os bancos de testes “PRINS” e “TUZUN” (Seção 5.2).

O algoritmo proposto foi implementado na linguagem de programação C++ utilizando o compilador GCC e foi executado em um Intel Core I7 com frequência de 2.96 GHz com 8GB de memória RAM e sistema operacional Ubuntu 64 bits Professional Edition. A parametrização para “MAX_ITER” é 16 vezes o número de depósitos disponíveis. Para “MAX_ILS1” é duas vezes o número de clientes. Por “MAX_ILS2” possuir perturbações mais leves e por ser executado poucas vezes foi parametrizado com 32 vezes o número de clientes vezes o número de depósitos.

Vários testes foram realizados nos três bancos para parametrização destes valores levando em conta o tempo de execução em relação as outras heurísticas, ou seja, as parametrizações procuram fazer com que o algoritmo não exceda proporcionalmente o tempo de execução, já que é executado em uma máquina mais rápida. O algoritmo ILS-RVND proposto pode obter resultados melhores que os apresentados se a parametrização for configurada para cada banco de teste. A expressão 5.1 apresenta como o *gap* utilizado nas tabelas foi calculado. Campo₁ e campo₂ são os valores a serem comparados. Exemplo: “gap/MRL” nas Tabelas 1 e 2 é o gap entre “MRL” (parâmetro₁) com “Melhor Solução do ILS-RVND”(parâmetro₂)

$$gap = 100 - \frac{(100 / \text{parâmetro}_1)}{\text{parâmetro}_2} \quad (5.1)$$

5.1 Comparação com as melhores soluções da literatura

Na Tabela 4, o campo “Instância” contém o nome da instância com o nome do Autor que propôs, ano de publicação, número de clientes e número de depósitos. Na Tabela 6, o campo “Instância” contém o nome da instância com o número de clientes, número de depósitos e versão. A versão “20-5-1a” se difere da “20-5-1b” pela capacidade dos veículos, capacidades dos depósitos e custo dos depósitos. Nas Tabelas 4, 5 e 6, o campo “MRL” representa os melhores resultados da literatura. O campo “Melhor Solução” é a melhor solução obtida pelo ILS-RVND em 10 execuções. “Média” é o custo médio das 10 execuções. Apresentam-se dois gaps: “gap/MRL” é o gap entre o campo “MRL” e o campo “Melhor Solução”; gap/Média é o gap entre o “MRL” e a “Média” do ILS-RVND. O campo “Tempo Médio” é o tempo médio das 10 execuções do algoritmo e estão expressos em segundos. Para facilitar a visualização, os resultados ótimos conhecidos são seguidos de asterisco, os resultados em negrito significam que empatam com os valores comparados e os resultados sublinhados são melhores que os resultados aos quais são comparados.

A Tabela 4 permite comparar os melhores resultados da literatura e o algoritmo ILS-RVND no primeiro banco de teste, utilizando o “gap/MRL” como parâmetro, de 19 instâncias, e considerando o valor $\pm 0,01$ como empate, evidenciando que este último apresentou três novas melhores soluções e empatou

em 13 resultados. A quantidade de empates se deve a grande parte das instâncias serem pequenas, com baixo número de clientes e depósitos facilitando sua resolução. Grande parte das soluções ótimas já foram encontradas para algumas instâncias deste banco. Pode-se notar que foram obtidas soluções bem melhores para as instâncias Perl83-318x4, ressaltando que outros autores realizam testes nesse banco, apenas Barreto et al (2007) possuem resultados publicados para estas instancias.

Tabela 4: Comparação das melhores soluções conhecidas com o ILS-RVND – Banco de Teste 1

Instância	Capacidade	MRL	ILS-RVND				
			Melhor Solução	Média	gap/MRL	gap/Média	Tempo Médio(s)
Christofides69-50x5	160	565,60*	565,60*	565,60*	0,00	0,00	4,24
Christofides69-75x10	140	802,08	845,93	848,96	5,18	5,52	26,99
Christofides69-100x10	200	833,40*	833,40*	834,20	0,00	0,10	110,32
Daskin95-88x8	9000000	355,80*	355,80*	355,80*	0,00	0,00	74,56
Daskin95-150x10	8000000	43919,90	43923,45	43975,78	0,01	0,13	374,80
Gaskell67-21x5	6000	424,90*	424,90*	424,90*	0,00	0,00	0,33
Gaskell67-22x5	4500	585,10*	585,11*	585,11*	0,00	0,00	0,49
Gaskell67-29x5	4500	512,10*	512,10*	512,10*	0,00	0,00	1,16
Gaskell67-32x5A	8000	562,22*	562,22*	562,22*	0,00	0,00	1,51
Gaskell67-32x5B	11000	504,30*	504,30*	504,30*	0,01	0,01	1,96
Gaskell67-36x5	250	460,40*	460,40*	460,40*	-0,01	-0,01	2,09
Min92-27x5	2500	3062,00*	3062,00*	3062,00*	0,00	0,00	0,91
Min92-134x8	850	5709,00	5719,25	5727,33	0,18	0,32	193,31
Perl83-12x2	140	204,00*	204,00	204,00	-0,01	-0,01	0,03
Perl83-55x15	120	1112,10	<u>1112,06</u>	1112,13	0,00	0,00	16,15
Perl83-85x7	160	1622,25	1622,50	1623,40	0,02	0,07	27,99
Perl83-318x4	25000	580680,20	<u>559808,84</u>	<u>565228,68</u>	-3,73	-2,73	3097,47
Perl83-318x4	8000	747619,00	<u>674392,12</u>	<u>678045,57</u>	-10,86	-10,26	1578,55
Or76-117x14	150	12474,20	<u>12296,93</u>	<u>12302,51</u>	-1,44	-1,40	332,58

Na Tabela 5, pode-se observar a comparação entre os melhores resultados da literatura e o algoritmo ILS-RVND para o segundo banco de teste, de 30 instâncias, quando se obteve cinco novas melhores soluções, empatando em 16 e perdendo em nove resultados. O campo utilizado como parâmetro foi o “gap/MRL”, considerando o valor $\pm 0,01$ como empate. Assim como no primeiro banco de teste, a quantidade de empates se deve a grande parte das instâncias serem pequenas, com baixo número de clientes e depósitos facilitando sua resolução. As instâncias “100-100-1a” e “100-10-1b” apresentam resultados ruins por possuírem uma carga apertada. Foi comprovado nos testes paralelos que aumentando o número de

construções o algoritmo possuirá maior probabilidade de encontrar melhores soluções nestas instâncias.

Tabela 5: Comparação das melhores soluções conhecidas com o ILS-RVND – Banco de Teste 2

Instância	MRL	ILS-RVND				Tempo Médio(s)
		Melhor Solução	Média	gap/MRL	gap/Média	
20-5-1 ^a	54793*	54793	54793	0,00	0,00	0,25
20-5-1b	39104*	39104	39104	0,00	0,00	0,38
20-5-2 ^a	48908*	48908	48908	0,00	0,00	0,24
20-5-2b	37542*	37542	37542	0,00	0,00	0,35
50-5-1 ^a	90111*	90111	90111	0,00	0,00	3,26
50-5-1b	63242*	63242	63247,6	0,00	0,01	5,10
50-5-2 ^a	88298*	88298	88298	0,00	0,00	3,73
50-5-2b	67308*	67308	67321	0,00	0,02	6,03
50-5-2BIS	84055*	84055	84055	0,00	0,00	5,60
50-5-2bBIS	51822*	51822	51870,8	0,00	0,09	3,25
50-5-3 ^a	86203*	86203	86203	0,00	0,00	3,03
50-5-3b	61830*	61830	61830	0,00	0,00	5,51
100-5-1 ^a	274814*	275281	275538,5	0,17	0,26	24,49
100-5-1b	213615	213654	214142,6	0,02	0,25	41,35
100-5-2	193671*	193671	193849,7	0,00	0,09	23,35
100-5-2b	157095*	157129	157162,3	0,02	0,04	41,73
100-5-3a	200079*	200131	200271,7	0,03	0,10	22,17
100-5-3b	152441*	152441	152573,6	0,00	0,09	32,66
100-10-1a	287983	303100	310848,2	4,99	7,36	45,66
100-10-1b	231763	237902	244982,1	2,58	5,40	65,95
100-10-2a	243590*	243590	243758,6	0,00	0,07	46,40
100-10-2b	203988	203988	204068,3	0,00	0,04	67,35
100-10-3a	250882	252992	259787,8	0,83	3,43	46,41
100-10-3b	204317	204661	204865,1	0,17	0,27	73,56
200-10-1a	477248	<u>476822</u>	496095,3	-0,09	3,80	400,51
200-10-1b	378065	<u>376982</u>	378448,5	-0,29	0,10	723,89
200-10-2a	449571	<u>448735</u>	<u>449094,2</u>	-0,19	-0,11	498,12
200-10-2b	374330	<u>373837</u>	374396	-0,13	0,02	677,51
200-10-3a	469433*	471147	484970,2	0,36	3,20	400,01
200-10-3b	362817	<u>362664</u>	363271,8	-0,04	0,13	669,42

Na Tabela 6, pode-se observar a comparação entre os melhores resultados da literatura e o algoritmo ILS-RVND para o terceiro banco de teste, de 36 instâncias, quando se obteve 19 novas melhores soluções, empatando em 11 e perdendo em sete resultados. O campo utilizado como parâmetro foi o “gap/MRL”, considerando o valor $\pm 0,01$ como empate. Por este banco de teste possuir instâncias com 100 a 200 clientes e 10 a 20 depósitos, o algoritmo ILS-RVND obteve grande vantagem ao ser aleatório à escolha dos depósitos.

Tabela 6: Comparação das melhores soluções conhecidas com o ILS-RVND – Banco de Teste 3

Instancia	MRL	ILS-RVND				
		Custo	Média	gap/MRL	gap/Média	Tempo Médio (s)
111112	1467,68*	1467,68*	1470,85	0,00	0,22	115,362
111122	1449,2	<u>1448,37</u>	1449,11	-0,06	-0,01	186,144
111212	1394,8*	1394,80*	1395,83	0,00	0,07	93,806
111222	1432,29	1432,29	1432,29	0,00	0,00	196,288
112112	1167,16*	1167,16*	1167,43	0,00	0,02	124,976
112122	1102,24	1102,24	1102,24	0,00	0,00	216,321
112212	791,66*	791,66*	792,11	0,00	0,06	113,712
112222	728,3	728,30	728,30	0,00	0,00	186,869
113112	1238,24	1238,49	1239,08	0,02	0,07	91,189
113122	1245,31	1245,31	1245,61	0,00	0,02	230,841
113212	902,26	916,05	916,05	1,51	1,51	107,358
113222	1018,28	1018,29	1018,29	0,00	0,00	203,546
131112	1866,75	1892,17	1917,64	1,34	2,65	324,474
131122	1833,95	<u>1820,32</u>	<u>1829,96</u>	-0,75	-0,22	768,106
131212	1965,12	1966,07	1971,01	0,05	0,30	381,722
131222	1801,39	<u>1794,83</u>	1803,18	-0,37	0,10	651,2
132112	1443,33*	1443,33*	1444,47	0,00	0,08	516,024
132122	1441,98	<u>1429,30</u>	<u>1437,86</u>	-0,89	-0,29	839,573
132212	1205,09	1208,54	1209,05	0,29	0,33	418,457
132222	930,99	<u>926,38</u>	940,32	-0,50	0,99	1077,943
133112	1699,92	<u>1694,18</u>	<u>1698,52</u>	-0,34	-0,08	372,951
133122	1400,01	<u>1392,01</u>	<u>1395,44</u>	-0,57	-0,33	876,3
133212	1199,51	1206,14	1206,56	0,55	0,58	463,543
133222	1152,18	<u>1151,80</u>	<u>1151,82</u>	-0,03	-0,03	836,431
121112	2259,87	<u>2239,36</u>	<u>2254,68</u>	-0,92	-0,23	941,96
121122	2185,41	<u>2139,64</u>	<u>2155,11</u>	-2,14	-1,41	1777,197
121212	2234,78	<u>2202,19</u>	<u>2211,54</u>	-1,48	-1,05	982,433
121222	2241,04	<u>2218,20</u>	<u>2227,09</u>	-1,03	-0,63	2037,61
122112	2089,77	<u>2073,86</u>	<u>2076,82</u>	-0,77	-0,62	1320,114
122122	1709,56	<u>1685,78</u>	<u>1693,71</u>	-1,41	-0,94	2565,887
122212	1466,62	<u>1449,46</u>	<u>1453,48</u>	-1,18	-0,90	946,358
122222	1084,78	<u>1082,46</u>	<u>1084,30</u>	-0,21	-0,04	1620,844
123112	1970,44	<u>1949,16</u>	<u>1955,59</u>	-1,09	-0,76	1137,912
123122	1918,93	1919,19	1926,14	0,01	0,37	1961,164
123212	1771,06	<u>1760,13</u>	<u>1762,16</u>	-0,62	-0,50	962,984
123222	1393,16	<u>1391,17</u>	1396,36	-0,14	0,23	1773,024

5.2 Comparação com o algoritmo ALNS

Nas Tabelas 7 e 8 são apresentados os resultados dos testes comparativos do algoritmo ILS-RVND com o ALNS proposto por Hemmelmayr et al.(2011) para os bancos de testes de “PRINS” e “TUZUN”. Nestas tabelas, o campo “Melhor Solução” apresenta o melhor resultado obtido em 5 execuções do algoritmo ALNS e em 10

execuções do algoritmo ILS-RVND. O campo “Média” apresentam o resultado médio das várias execuções dos algoritmos para as instâncias. O campo “Tempo Médio” é o tempo médio das várias execuções para cada instância. O “gap/MS” é o *gap* entre as melhores soluções de cada algoritmo. O campo “gap/Média” é o *gap* entre as médias das soluções dos dois algoritmos.

Tabela 7: Comparação entre os algoritmos ALNS e ILS-RVND – Banco de Teste 2

Instância	ALNS			ILS-RVND			gap/MS	gap/Média
	Melhor Solução	Média	Tempo Médio(s)	Melhor Solução	Média	Tempo Médio(s)		
20-5-1a	54793	54793,00	39,00	54793	54793	0,25	0,00	0,00
20-5-1b	39104	39104,00	54,00	39104	39104	0,38	0,00	0,00
20-5-2a	48908	48908,00	38,00	48908	48908	0,24	0,00	0,00
20-5-2b	37542	37542,00	67,00	37542	37542	0,35	0,00	0,00
50-5-1	90111	90111,00	101,00	90111	90111	3,26	0,00	0,00
50-5-1b	63242	63242,00	65,00	63242	63247,6	5,10	0,00	0,01
50-5-2	88443	88576,80	99,00	<u>88298</u>	<u>88298</u>	3,73	-0,16	-0,32
50-5-2b	67340	67448,20	200,00	<u>67308</u>	<u>67321</u>	6,03	-0,05	-0,19
50-5-2bis	84055	84119,00	107,00	84055	<u>84055</u>	5,60	0,00	-0,08
50-5-2bbis	51822	51840,00	98,00	51822	51870,8	3,25	0,00	0,06
50-5-3	86203	86261,60	101,00	86203	<u>86203</u>	3,03	0,00	-0,07
50-5-3b	61830	61830,00	137,00	61830	61830	5,51	0,00	0,00
100-5-1	275636	276364,00	520,00	<u>275281</u>	<u>275538,5</u>	24,49	-0,13	-0,30
100-5-1b	214735	215059,00	1190,00	<u>213654</u>	<u>214142,6</u>	41,35	-0,51	-0,43
100-5-2	193752	193903,00	463,00	<u>193671</u>	<u>193849,7</u>	23,35	-0,04	-0,03
100-5-2b	157095	157156,60	859,00	157129	157162,3	41,73	0,02	0,00
100-5-3	200305	200495,60	454,00	<u>200131</u>	<u>200271,7</u>	22,17	-0,09	-0,11
100-5-3b	152441	152899,80	684,00	152441	<u>152573,6</u>	32,66	0,00	-0,21
100-10-1	296877	299982,40	210,00	303100	310848,2	45,66	2,05	3,50
100-10-1b	235849	240289,20	188,00	237902	244982,1	65,95	0,86	1,92
100-10-2	244740	245548,20	136,00	<u>243590</u>	<u>243758,6</u>	46,40	-0,47	-0,73
100-10-2b	204016	204494,60	261,00	<u>203988</u>	<u>204068,3</u>	67,35	-0,01	-0,21
100-10-3	253801	254882,00	202,00	<u>252992</u>	259787,8	46,41	-0,32	1,89
100-10-3b	205609	206175,20	224,00	<u>204661</u>	<u>204865,1</u>	73,56	-0,46	-0,64
200-10-1	480883	483204,80	752,00	<u>476822</u>	496095,3	400,51	-0,85	2,60
200-10-1b	378961	380538,80	1346,00	<u>376982</u>	<u>378448,5</u>	723,89	-0,52	-0,55
200-10-2	450451	451750,60	1201,00	<u>448735</u>	<u>449094,2</u>	498,12	-0,38	-0,59
200-10-2b	374751	376111,80	1349,00	<u>373837</u>	<u>374396</u>	677,51	-0,24	-0,46
200-10-3	475373	479366,60	1251,00	<u>471147</u>	484970,2	400,01	-0,90	1,16
200-10-3b	366902	369614,00	1137,00	<u>362664</u>	<u>363271,8</u>	669,42	-1,17	-1,75

Ao se comparar ainda os resultados obtidos pelos algoritmos ALNS (Hemmelmayr, 2011) e ILS-RVND (Tabela 7) ressaltou-se que, neste último algoritmo, 15 instâncias apresentaram melhores soluções, empataram em 12 e perderam em duas, quando utilizado o “gap/MS”. Como pode-se observar, ao comparar com um algoritmo específico, a qualidade do algoritmo ILS-RVND fica mais destacada, onde foi possível encontrar 15 melhores soluções em tempo

competitivo, ressaltando que os testes do algoritmo ALNS foi executado em um processador Opteron 275 AMD com frequência de 2.2 GHz contra o Intel Core i7 com frequência de 2.96 GHz do ILS-RVND.

Tabela 8: Comparação entre os algoritmos ALNS e ILS-RVND – Banco de Teste 3

Instancia	ALNS			ILS-RVND			gap/MS	gap/Média
	Melhor Solução	Média	Tempo Médio	Melhor Solução	Média	Tempo Médio		
111112	1467,68*	1475,67	275,00	1467,68*	<u>1470,85</u>	115,36	0,00	-0,33
111122	1452,14	1464,72	321,00	<u>1448,37</u>	<u>1449,11</u>	186,14	-0,26	-1,08
111212	1394,93	1400,49	244,00	1394,80*	<u>1395,83</u>	93,81	-0,01	-0,33
111222	1433,42	1441,21	376,00	<u>1432,29</u>	<u>1432,29</u>	196,29	-0,08	-0,62
112112	1167,53	1173,04	489,00	1167,16*	<u>1167,43</u>	124,98	-0,03	-0,48
112122	1102,24	1102,34	373,00	1102,24	<u>1102,24</u>	216,32	0,00	-0,01
112212	791,66*	791,83	739,00	791,66*	792,11	113,71	0,00	0,04
112222	728,30	728,32	384,00	728,30	728,30	186,87	0,00	0,00
113112	1238,70	1240,31	357,00	<u>1238,49</u>	<u>1239,08</u>	91,19	-0,02	-0,10
113122	1246,52	1248,17	445,00	<u>1245,31</u>	<u>1245,61</u>	230,84	-0,10	-0,21
113212	902,26	902,27	321,00	916,05	916,05	107,36	1,51	1,50
113222	1018,29	1018,56	386,00	1018,29	<u>1018,29</u>	203,55	0,00	-0,03
131112	1922,70	1939,52	504,00	<u>1892,17</u>	<u>1917,64</u>	324,47	-1,61	-1,14
131122	1847,93	1857,29	635,00	<u>1820,32</u>	<u>1829,96</u>	768,11	-1,52	-1,49
131212	1975,83	2009,44	664,00	<u>1966,07</u>	<u>1971,01</u>	381,72	-0,50	-1,95
131222	1806,31	1838,51	485,00	<u>1794,83</u>	<u>1803,18</u>	651,20	-0,64	-1,96
132112	1447,43	1449,15	1049,00	1443,33*	<u>1444,47</u>	516,02	-0,28	-0,32
132122	1445,32	1446,91	805,00	<u>1429,30</u>	<u>1437,86</u>	839,57	-1,12	-0,63
132212	1204,98	1205,83	2197,00	1208,54	1209,05	418,46	0,29	0,27
132222	931,49	933,14	982,00	<u>926,38</u>	940,32	1077,94	-0,55	0,76
133112	1694,64	1700,39	1046,00	<u>1694,18</u>	<u>1698,52</u>	372,95	-0,03	-0,11
133122	1400,50	1403,50	925,00	<u>1392,01</u>	<u>1395,44</u>	876,30	-0,61	-0,58
133212	1198,67	1199,27	1375,00	1206,14	1206,56	463,54	0,62	0,60
133222	1152,01	1154,36	911,00	<u>1151,80</u>	<u>1151,82</u>	836,43	-0,02	-0,22
121112	2265,15	2278,27	944,00	<u>2239,36</u>	<u>2254,68</u>	941,96	-1,15	-1,05
121122	2183,05	2192,61	847,00	<u>2139,64</u>	<u>2155,11</u>	1777,20	-2,03	-1,74
121212	2233,55	2247,75	907,00	<u>2202,19</u>	<u>2211,54</u>	982,43	-1,42	-1,64
121222	2230,94	2263,20	860,00	<u>2218,20</u>	<u>2227,09</u>	2037,61	-0,57	-1,62
122112	2082,60	2093,78	1606,00	<u>2073,86</u>	<u>2076,82</u>	1320,11	-0,42	-0,82
122122	1710,67	1732,00	941,00	<u>1685,78</u>	<u>1693,71</u>	2565,89	-1,48	-2,26
122212	1458,55	1462,15	1861,00	<u>1449,46</u>	<u>1453,48</u>	946,36	-0,63	-0,60
122222	1085,29	1086,08	812,00	<u>1082,46</u>	<u>1084,30</u>	1620,84	-0,26	-0,16
123112	1964,75	1971,01	968,00	<u>1949,16</u>	<u>1955,59</u>	1137,91	-0,80	-0,79
123122	1926,64	1952,31	740,00	<u>1919,19</u>	<u>1926,14</u>	1961,16	-0,39	-1,36
123212	1762,09	1764,16	2055,00	<u>1760,13</u>	<u>1762,16</u>	962,98	-0,11	-0,11
123222	1393,06	1395,38	1038,00	<u>1391,17</u>	1396,36	1773,02	-0,14	0,07

Ao se comparar os resultados obtidos pelos algoritmos ALNS (Hemmelmayr, 2011) e ILS-RVND (Tabela 8) para o terceiro banco de teste ressaltou-se que, neste último algoritmo, 27 instâncias apresentaram melhores soluções, empataram em 6 e perderam em três, quando utilizado o “gap/MS”. Neste

banco de teste o algoritmo ILS-RVND demonstra superioridade para encontrar melhores soluções.

5.3 Resumo

A heurística proposta ILS-RVND obteve resultados satisfatórios. Usando a média dos *gaps* das melhores soluções comparadas com as melhores soluções da literatura como critério, o algoritmo obteve -0,56 para o banco de teste “BARRETO”, 0,28 para “PRINS” e -0,30 para “TUZUN”. Novas soluções foram encontradas para os três bancos de testes e quando comparado com algoritmo ALNS, proposto por Hemmelmayr (2011), a média do “gap/MS” é -0,11 no banco de teste “PRINS” e -0,40 para “TUZUN”. O tempo de processamento comparado com o algoritmo ALNS foi menor para várias instâncias, porém não podemos afirmar que o algoritmo é mais rápido porque os códigos foram executados em diferentes máquinas. Apesar do algoritmo ter melhorado com as construções do tipo *First Fit Decreasing* construído para uma instância do primeiro banco de teste e várias do segundo, ele ainda apresenta dificuldades para encontrar as mesmas soluções que outros algoritmos da literatura. Para as instâncias maiores, durante o teste paralelo, comprovou-se que o algoritmo ainda possui poder para encontrar melhores soluções a custo computacional, ou seja, aumentando o número de iterações. Em vista destes resultados, no próximo capítulo são propostas versões distribuídas do algoritmo com a finalidade de explorar todo o potencial do algoritmo.

Capítulo 6 – Versão Paralela

O PLR é um problema de alta complexidade, sendo sua versão de decisão definida como problema NP-Completo por Barreto (2004). Como consequência, é exigido um grande esforço computacional encontrar uma solução ótima a media que as instâncias aumentam de tamanho. Duas abordagens são comumente utilizadas: as exatas e as heurísticas. Na literatura, nenhum método exato resolveu problemas com mais de 100 nós sem utilizar grandes simplificações do problema. Em contrapartida, as meta-heurísticas ganham espaço e produzem soluções de boa qualidade em um tempo computacional aceitável para o PLR.

Nos últimos anos, em vista à facilidade de acesso a arquiteturas com multiprocessadores, GPU e *cluster*, vários métodos tem sido propostos para paralelizar as meta-heurísticas para os mais diversos problemas de otimização (ALBA, 2012). A paralelização toma vantagem do grande poder de processamento disponível e ajuda os algoritmos as explorarem melhor o espaço de busca em um tempo aceitável. Alba et al. (2012) propõem uma divisão dos modelos de paralelização para meta-heurísticas baseadas em busca-local em três metodologias:

- *The parallel exploration and evaluation of the neighborhood*: é um método mestre-escravo de baixo nível onde uma busca sequencial deve computar o mesmo resultado. No início de cada iteração, o mestre replica a solução corrente entre os nós distribuídos. Cada um administra separadamente suas soluções e resultados e retornam ao mestre (Figura 9a).
- *The parallel multistart model*: se baseia em simultaneamente disparar várias meta-heurísticas de busca-local para computar soluções melhores. Os algoritmos devem ser heterogêneos ou homogêneos, independentes ou cooperativos, começando das mesmas ou diferentes soluções e configurados com os mesmos ou diferentes parâmetros (Figura 9b).
- *The parallel evaluation of a single solution*: A qualidade de cada movimento é avaliada de maneira centralizada. Este modelo é particularmente interessante quando a função de avaliação pode ser paralelizada. Neste caso, a função

pode ser vista como uma agregação de um certo número de funções parciais que podem ser executadas em paralelo (Figura 9c).

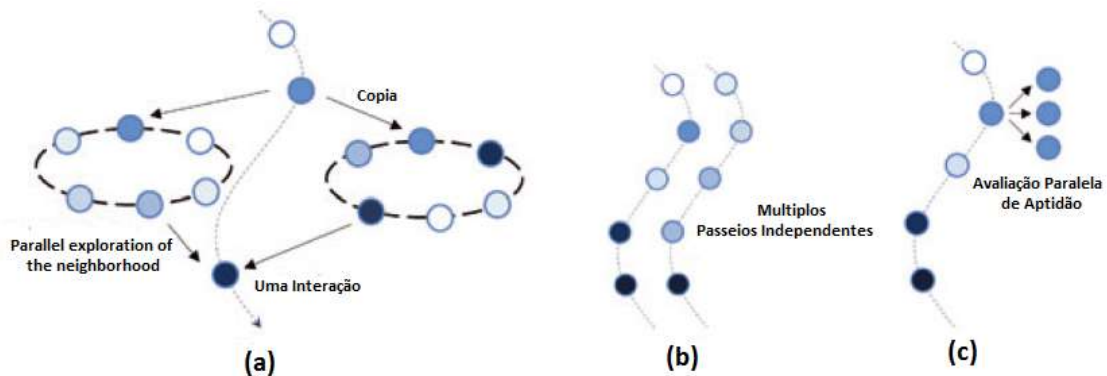


Figura 9: As três classificações proposta por Alba et al para meta-heurísticas baseadas em busca-local. (a) *Parallel exploration and evaluation of the neighborhood*, (b) *parallel multistart model* e (c) *parallel evaluation of a single solution* (Alba, 2012).

6.1 Versões Paralelas

O trabalho de paralelizar o algoritmo proposto ILS-RVND foi incentivado pelos resultados obtidos na versão sequencial. Foi verificado que se o algoritmo fosse configurado direcionado a um banco de teste ou aumentasse o tempo de execução do algoritmo, os resultados melhorariam consideravelmente. O trabalho de paralelização do algoritmo consistiu inicialmente em distribuí-lo de maneira a não modificar a hierarquia dos módulos do algoritmo. Inicialmente, duas versões principais do algoritmo foram construídas com o objetivo de distribuir o trabalho realizado pela versão sequencial. Assim, a ideia é dividir o trabalho realizado pelo algoritmo sequencial em N linhas de execução. A principal diferença entre estas versões consiste na existência de comunicação, ou não, entre as diversas linhas de execução. Para melhor explicarmos as versões, dividiremos o algoritmo 2 apresentado, no Capítulo 4, em duas fases:

- Fase 1: construção e refinamento 1 (linhas 5 a 12 do algoritmo 2);
- Fase 2: refinamento 2 (linhas 13 a 17 do algoritmo 2).

Para distribuição do trabalho, os parâmetros “MAX_ITER” e “MAX_ILS2” foram divididos por N e estas iterações foram distribuídas em N linhas de execução

(*threads*), considerando a existência de N processadores disponíveis. Como o parâmetro “MAX_ILS1” está dentro do laço de parâmetro “MAX_ITER”, o seu valor continua o mesmo do sequencial. O Algoritmo é apresentado novamente a seguir:

Algoritmo 2. ILS-RVND(Instancia)

```

1.  frota ← Estima_Frota(Instancia)
2.  depositosEstimados ← Estima_Depositos(Instancia)
3.  s, s', s'' ← 0
4.  Para depositosEstimados < Instancia.Depositos ou rodada sem melhora faça
5.      Para i ← 0 até i < MAX_ITER faça
6.          s ← Construcão(frota, depositosEstimados);
7.          s ← RVND(s);
8.          Para j ← 0 até j < MAX_ILS1 faça
9.              s' ← Perturbacao1(s, histórico);
10.             s'' ← RVND(s');
11.             s ← CriterioAceitacao(s,s'', histórico)
12.             Se (s==s'') faça j ← 0
13.         Para j ← 0 até j < MAX_ILS2 faça
14.             s' ← Perturbacao2(s, histórico);
15.             s'' ← RVND(s');
16.             s ← CriterioAceitacao(s,s'', histórico)
17.             Se (s==s'') faça j ← 0
18.  Retorna(s)

```

6.1.1 Versão sem comunicação

Nesta primeira versão, denotada *vDistrib*, o trabalho do algoritmo sequencial é distribuído igualmente entre as *threads* (linhas de execução). Neste caso, não existe comunicação entre elas, que irão trabalhar independentemente uma das outras. Ao final da execução das *threads*, é escolhida a melhor solução encontrada.

A criação desta versão foi motivada pelo desejo de se diminuir o tempo total de execução na busca de uma solução para o problema. Neste caso, o objetivo inicial não foi encontrar melhores soluções, mas sim manter o mesmo padrão de qualidade de soluções porém em um tempo computacional menor. O ideal é que se o problema for dividido em N partes, seja possível encontrar uma solução equivalente em um tempo N vezes menor.

Usando as classificações propostas por Alba et al. (2012), pode-se classificar esta primeira versão como um *parallel multistart model* homogêneo, ou seja, os algoritmos são homogêneos, independentes, começam de diferentes soluções e configurados com os mesmos parâmetros. Na Figura 10 podemos visualizar um esquema gráfico da primeira versão de distribuição do trabalho utilizando quatro *threads*.

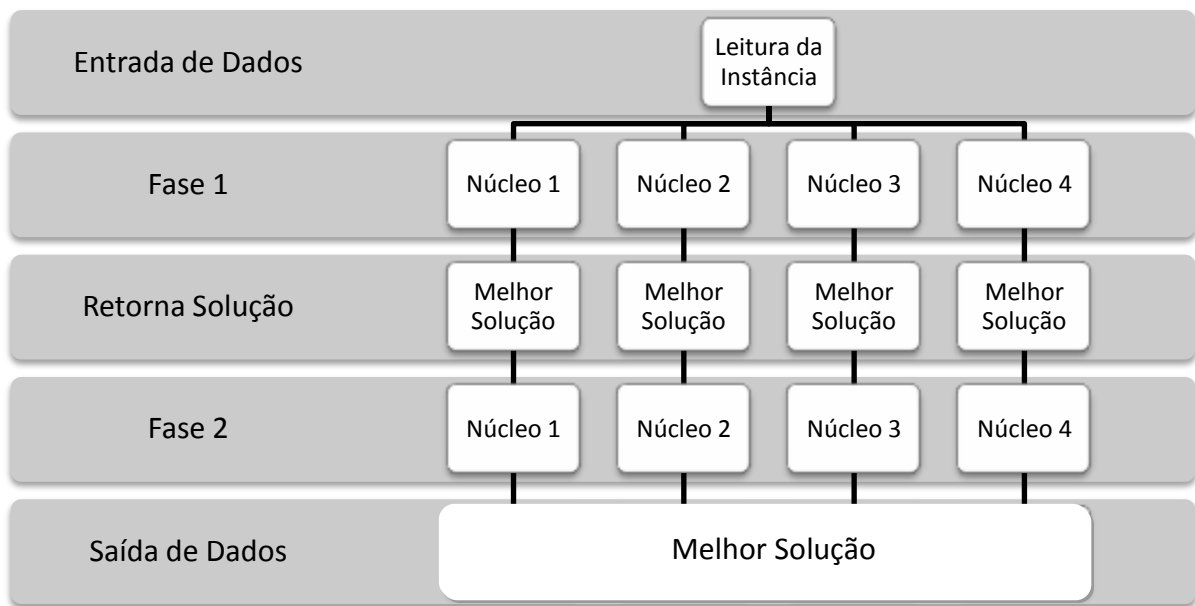


Figura 10: Distribuição de trabalho da versão *vDistrib*

6.1.2 Versão com Comunicação

Com a intenção de encontrar melhores soluções, nesta segunda versão chamada de *vParallel*, as *threads* retornam a melhor solução obtida na primeira fase do algoritmo para a fase dois de todas as *threads*. Assim, diferentemente da primeira versão (*vDistrib*), os refinadores da fase 2 não utilizarão soluções diferentes. Outra modificação importante na segunda fase é que quando a solução é atualizada, todas as *threads* têm acesso à nova solução através da memória compartilhada. Em caso de atualização da melhor solução, o contador de iterações dos refinamentos é zerado em todas as *threads*, evitando que o refinador pare previamente em comparação à versão sequencial.

Diferentemente da versão *vDistrib*, a classificação de *parallel multistart model* homogêneo se mantém apenas para a primeira fase. Na segunda fase, fica configurado um *parallel exploration and evaluation of the neighborhood*, porque além

de trabalhar com a mesma solução no início da segunda fase, toda atualização de uma melhor solução é replicada para os outros *threads*. Na Figura 11, podemos visualizar um esquema gráfico da versão *vParallel* e observar que o compartilhamento das soluções encontradas entre as *threads* é feito através de memória compartilhada. Observar que não há uma sincronização entre os processadores ao final de cada fase. Quando uma *thread* termina a fase 1, a fase 2 é iniciada com a melhor solução conhecida até o momento naquela máquina.

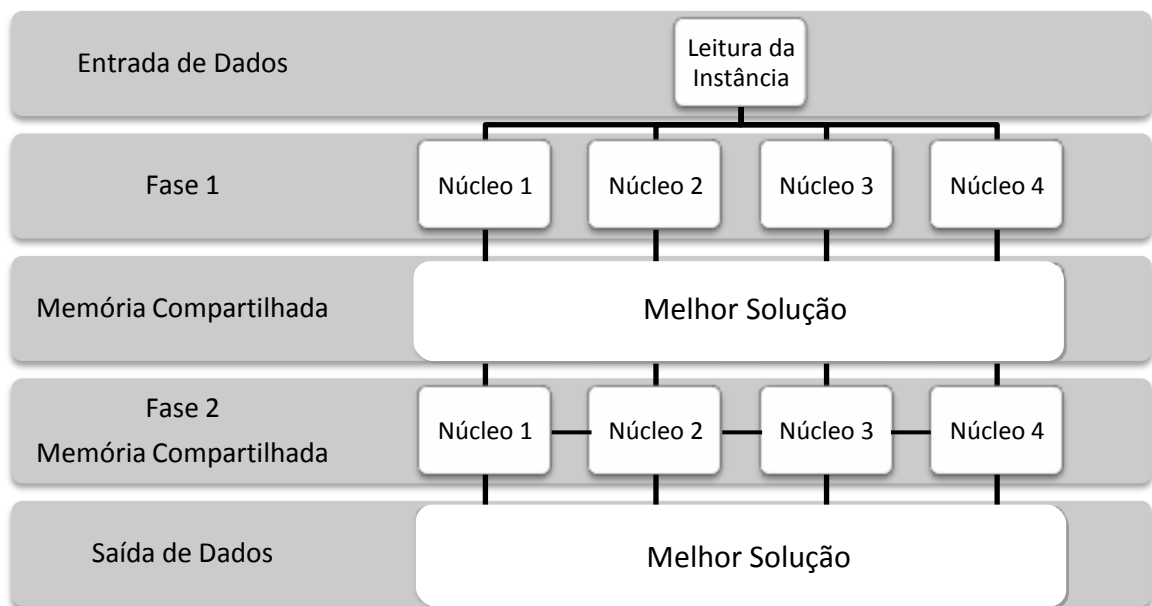


Figura 11: Distribuição de trabalho e comunicação da versão *vParallel*

Como já destacado, a comunicação na versão *vParallel* é feita através da memória compartilhada. Neste caso, a comunicação acontece por exemplo entre as *threads* que executam nos núcleos (*cores*) de uma mesma máquina. Para que a comunicação aconteça entre diferentes máquinas com memória distribuída, foi utilizada a biblioteca MPI para troca de mensagens. Esta versão com comunicação em memória distribuída será denotada *vParallel_MPI*.

Na versão *vParallel_MPI*, é adicionada uma *thread* por máquina, sendo esta usada apenas para comunicação. Esta *thread* é responsável por toda vez que ocorrer uma atualização na memória compartilhada, replicar a solução para os outros processadores. A quantidade de mensagens trocadas entre diferentes máquinas é baixa.

Na Figura 12 é possível visualizar um esquema gráfico da versão *vParallel_MPI*. Na figura, são apresentadas duas máquinas e quatro núcleos. A

comunicação entre o comunicador e os núcleos é feita através de memória compartilhada enquanto entre diferentes máquinas é realizada pelos comunicadores através da biblioteca MPI. Assim como na versão *vParallel*, não há uma sincronização entre os processadores ao final de cada fase. Quando uma *thread* termina a fase 1, a fase 2 é iniciada, porém, nesta versão, a *thread* tem acesso a melhor solução conhecida até o momento em todas as máquinas.

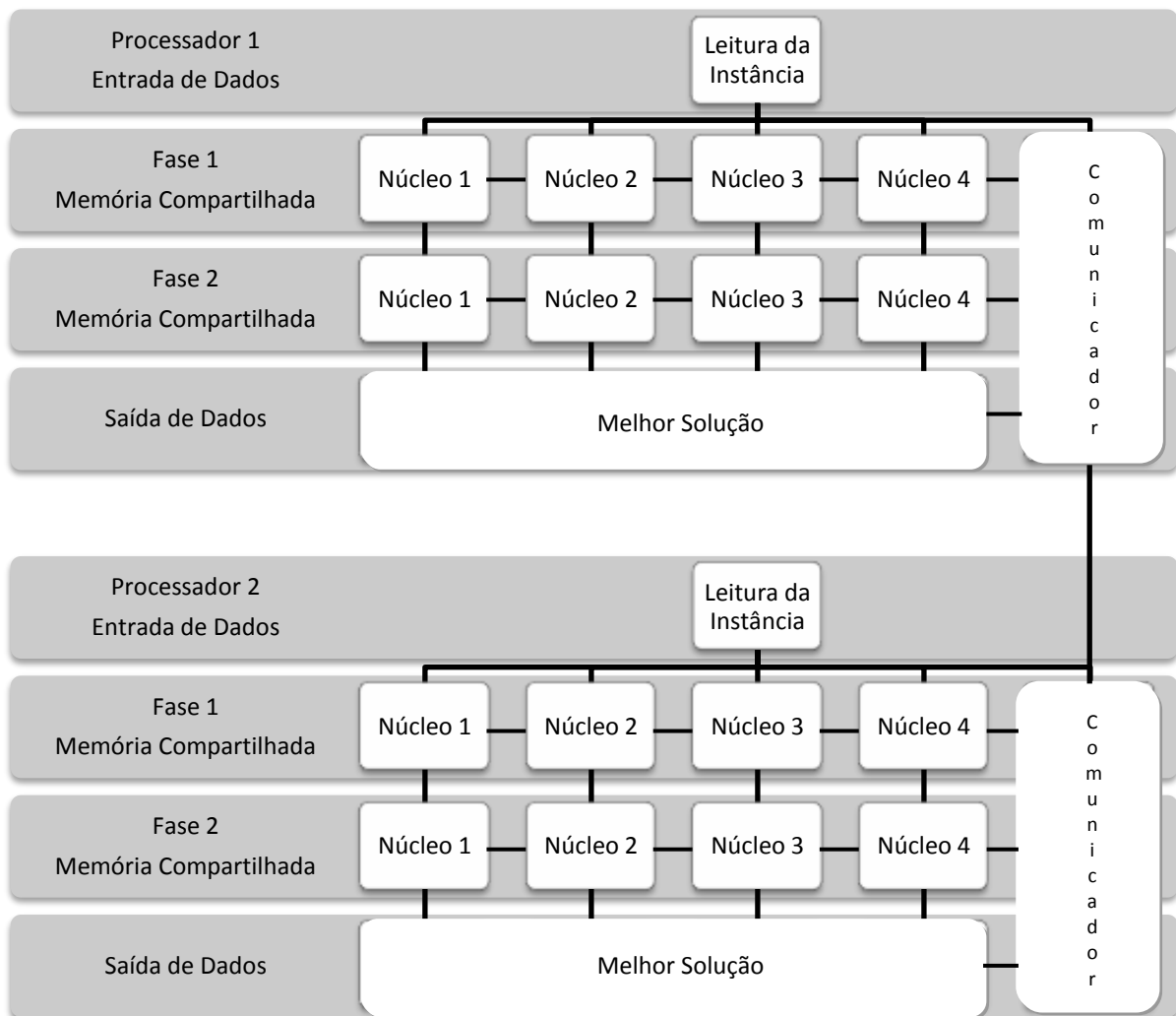


Figura 12: Distribuição de trabalho e comunicação da versão *vParallel_MPI*

6.2 Análise Experimental: *vDistrib* × *vParallel* × *vParallel_MPI*

Assim como na versão sequencial, as versões paralelas do algoritmo foram preliminarmente testadas considerando os mesmos bancos de testes. Estas versões foram implementadas na linguagem de programação C++ utilizando o compilador GCC.

6.2.1 Análise Inicial: Cluster LAPORTE

Os testes foram executados em quatro máquinas, cada uma com um processador Intel Core 2 Quad Q8200 com 2.33 GHz e 1,98GB. Assim, cada máquina possui 4 núcleos, totalizando no máximo 16 núcleos de execução para este experimento. O sistema operacional utilizado foi o Ubuntu 64-bit, versão 12.0.4. Adicionalmente, a biblioteca POSIX *Pthread* foi utilizada para programar as várias linhas de execução (*threads*) e a comunicação entre máquinas foi realizada através da biblioteca MPI. Nesta primeira bateria de testes, comparam-se as melhores soluções encontradas por *vDistrib*, *vParallel* e *vParallel_MPI* e a média dos resultados em dez execuções com as melhores soluções encontradas pela versão sequencial, com a finalidade de comparar uma relação entre *speedup* e qualidade da solução. Os números de *threads* utilizados foram de 1, 2, 4, 8 e 16. Observar que entre *vParallel* e *vParallel_MPI* os resultados até quatro *threads* são os mesmos, por não haver comunicação entre nós, apenas comunicação via memória compartilhada.

Nestes testes, a parametrização do campo “MAX_ITER” na versão sequencial é 16 vezes o número de depósitos disponíveis. Nas versões paralelas, este valor é distribuído de acordo com o número de *threads*. Por exemplo, para duas *threads*, cada execução ficaria com “MAX_ITER” igual a oito vezes o número de depósitos; quatro vezes o número de depósitos para quatro *threads*; e duas vezes para oito *threads*. Para o campo “MAX_ILS1”, a parametrização é duas vezes o número de clientes na versão sequencial e paralela, já que o número de execuções dessa região do código é sub-rotina do campo controlado por “MAX_ITER”. Na versão sequencial, o campo “MAX_ILS2” foi parametrizado com 32 vezes o número de clientes vezes o número de depósitos. Seguindo a lógica do campo “MAX_ITER”, este valor foi distribuído de acordo com o número de *threads*. Por exemplo, para duas *threads*, cada execução ficaria com “MAX_ILS2” igual a 16 vezes o número de clientes vezes o número de depósitos; oito vezes o número de clientes vezes o número de depósitos para quatro *threads*; e quatro vezes para oito *threads*.

Nas Tabelas 9 a 29, apresentadas neste capítulo, o campo “Instância” apresenta o nome da instância como foi explicado no Capítulo 5 para os bancos de testes 1, 2 e 3. O campo “Custo” é a melhor solução obtida pelo ILS-RVND em 10 execuções. “Média” é o custo médio das 10 execuções. Apresentam-se dois *gaps*: “gap/MS” é o *gap* entre o campo “Custo” da versão sequencial e o campo “Custo”

da versão paralela; “gap/Média” é o *gap* entre os campos “Média” da versão sequencial e paralela. O campo “Tempo Médio” é o tempo médio das 10 execuções do algoritmo e está expresso em segundos. O campo “*SpeedUp*” é calculado através da divisão do “Tempo Médio” da versão sequencial pelo “Tempo Médio” da versão paralela. Para facilitar a visualização, os resultados ótimos conhecidos estão em negrito seguidos de asterisco, os resultados em negrito significam que empatam com os valores comparados e os resultados sublinhados são melhores que os resultados aos quais são comparados.

Para melhor analisar os valores de *speedups* obtidos nas versões paralelas, testes revelam que, na versão sequencial, as construções não gastam em média nem 0,1% do tempo de execução, 70% a fase 1 e 30% a fase 2.

As Tabelas 9 a 17 mostram os resultados comparativos obtidos para o algoritmo sequencial e as versões com 16 *threads*. Os demais resultados para 2, 4 e 8 *threads* são apresentados resumidamente nos gráficos das Figuras 13 a 18.

A Tabela 9 permite comparar a versão *vDistrib*, executada com 16 *threads*, com a versão sequencial. A versão *vDistrib*, utilizando o “gap/MS” como parâmetro, de 19 instâncias, apresentou uma melhora e empatou em 16 resultados. O *speedup*

Tabela 9: Comparação do sequencial com *vDistrib* – Banco de Teste 1

Instâncias	Sequencial			vDistrib – 16 Threads			gap/ MSC	gap/ Média	Speedup
	Custo	Média	Tempo	Custo	Média	Tempo			
Christofides69-50x5	565,6*	565,6*	6,16	565,6*	565,6*	0,66	0,00	0,00	9,36
Christofides69-75x10	844,6	848,9	37,99	844,6	849,5	5,53	0,00	0,07	6,87
Christofides69-100x10	833,4*	834,6	146,07	833,4*	<u>833,4</u>	18,12	0,00	-0,14	8,06
Daskin95-88x8	355,8	355,8	100,83	355,8	355,8	10,07	0,00	0,00	10,01
Daskin95-150x10	43919,9	43990,4	536,32	43919,9	<u>43943,2</u>	74,52	0,00	-0,11	7,20
Gaskell67-21x5	424,9*	424,9	0,46	424,9*	424,9*	0,07	0,00	0,00	6,83
Gaskell67-22x5	585,1*	585,1	0,71	585,1*	585,1*	0,08	0,00	0,00	9,31
Gaskell67-29x5	512,1*	512,1	1,66	512,1*	512,1*	0,15	0,00	0,00	10,86
Gaskell67-32x5A	562,2*	562,2	1,93	562,2*	562,2*	0,19	0,00	0,00	10,13
Gaskell67-32x5B	504,3*	504,3	2,29	504,3*	504,3*	0,20	0,00	0,00	11,62
Gaskell67-36x5	460,4*	460,4	2,86	460,4*	460,4*	0,27	0,00	0,00	10,68
Min92-27x5	3062,0*	3062,0	1,34	3062,0*	3062,0*	0,14	0,00	0,00	9,80
Min92-134x8	5723,6	5751,3	271,97	<u>5719,3</u>	<u>5732,5</u>	40,13	-0,08	-0,33	6,78
Perl83-12x2	204,0*	204,0	0,04	204,0*	204,0*	0,01	0,00	0,00	4,65
Perl83-55x15	1112,1*	1112,3	23,92	1112,1*	<u>1112,1</u>	3,65	0,00	-0,02	6,56
Perl83-85x7	1622,5*	1623,7	39,58	1622,5	<u>1623,4</u>	6,13	0,00	-0,02	6,45
Perl83-318x4	558680,3	565035,4	4118,51	560292,3	<u>564611,8</u>	548,04	0,29	-0,08	7,51
Perl83-318x4	670804,5	678144,5	2099,07	672400,4	681782,2	563,42	0,24	0,53	3,73
Or76-117x14	12296,9	12302,5	449,12	12296,9	<u>12300,6</u>	41,67	0,00	-0,02	10,78

obteve a média de 8,27, oscilando entre os valores de 3,73 à 11,62. A versão valores de “gap/MSC” compatíveis com a versão sequencial, com média de 0,02, conseguindo melhorar consideravelmente os valores de “gap/Média” de 7 instâncias obtendo um valor médio de -0,01.

Na Tabela 10, encontra-se os resultados obtidos pela comparação das versão *vParallel*, utilizando 16 *threads*, com a versão sequencial para o primeiro banco de teste. Utilizando o “gap/MSC” como parâmetro, a versão *vParallel* obteve uma melhora e empatou em 15 resultados de 19 instâncias. O *speedup* obteve a média de 6,36, oscilando entre os valores de 2,49 à 9,36. O resultados obtidos mostram que esta versão não conseguiu melhorar os resultados no parâmetro “gap/MSC” nem “gap/Média” em relação a versão sequencial, porém seus resultados não foram ruins, não passando na média dos valores 0,09 para o “gap/MSC” e 0,06 para o parâmetro “gap/Média”. Esta oscilação de valores pode ser encontrada quando realizam-se várias execuções diferentes da versão sequencial.

Tabela 10: Comparação do sequencial com *vParallel* – Banco de Teste 1

Instâncias	Sequencial			vParallel – 16 Threads			gap/ MSC	gap/ Média	Speedup
	Custo	Média	Tempo	Custo	Média	Tempo			
Christofides69-50x5	565,6*	565,6*	6,16	565,6*	565,6*	0,85	0,00	0,00	7,21
Christofides69-75x10	844,6	848,9	37,99	845,9	851,6	7,02	0,16	0,32	5,41
Christofides69-100x10	833,4*	834,6	146,07	833,4*	835,3	21,24	0,00	0,08	6,88
Daskin95-88x8	355,8	355,8	100,83	355,8	355,8	12,32	0,00	0,00	8,18
Daskin95-150x10	43919,9	43990,4	536,32	43919,9	44031,6	84,87	0,00	0,09	6,32
Gaskell67-21x5	424,9*	424,9	0,46	424,9*	424,9*	0,11	0,00	0,00	4,32
Gaskell67-22x5	585,1*	585,1	0,71	585,1*	585,1*	0,11	0,00	0,00	6,26
Gaskell67-29x5	512,1*	512,1	1,66	512,1*	512,1*	0,23	0,00	0,00	7,28
Gaskell67-32x5A	562,2*	562,2	1,93	562,2*	562,2*	0,27	0,00	0,00	7,19
Gaskell67-32x5B	504,3*	504,3	2,29	504,3*	504,3*	0,27	0,00	0,00	8,36
Gaskell67-36x5	460,4*	460,4	2,86	460,4*	460,4*	0,39	0,00	0,00	7,31
Min92-27x5	3062,0*	3062,0	1,34	3062,0*	3062,0*	0,20	0,00	0,00	6,72
Min92-134x8	5723,6	5751,3	271,97	<u>5719,8</u>	5751,9	44,82	-0,07	0,01	6,07
Perl83-12x2	204,0*	204,0	0,04	204,0*	204,0*	0,02	0,00	0,00	2,49
Perl83-55x15	1112,1*	1112,3	23,92	1112,1*	<u>1112,1*</u>	4,16	0,00	-0,02	5,76
Perl83-85x7	1622,5*	1623,7	39,58	1622,5*	<u>1623,1</u>	7,08	0,00	-0,04	5,59
Perl83-318x4	558680,3	565035,4	4118,51	561877,3	<u>564802,3</u>	601,67	0,57	-0,04	6,85
Perl83-318x4	670804,5	678144,5	2099,07	678278,6	682407,5	625,79	1,10	0,62	3,35
Or76-117x14	12296,9	12302,5	449,12	12296,9	12307,0	48,01	0,00	0,04	9,36

Analisando a Tabela 11, que compara a versão sequencial com a *vParallel_MPI* para o primeiro banco de teste e utiliza 16 *threads*, verifica-se que, no parâmetro “gap/MSC”, de 19 instâncias, obteve-se uma melhora e 15 empates com

speedup médio de 4,96, com valores de 0,16 a 10,45. Assim como a versão *vParallel*, os resultados obtidos não apresentam melhora na média geral para os parâmetros “gap/MS” e “gap/Média” em relação a versão sequencial. Seus resultados foram apenas um pouco melhores que *vParallel*, sendo a média dos valores 0,07 para o “gap/MS” e 0,05 para o parâmetro “gap/Média”.

Tabela 11: Comparação do sequencial com *vParallel_MPI* – Banco de Teste 1

Instâncias	Sequencial			<i>vParallel_MPI</i> – 16 Threads			gap/ MS	gap/ Média	Speedup
	Custo	Média	Tempo	Custo	Média	Tempo			
Christofides69-50x5	565,6*	565,6*	6,16	565,6*	566,4	1,21	0,00	0,14	5,10
Christofides69-75x10	844,6	848,9	37,99	845,9	849,9	5,98	0,15	0,12	6,35
Christofides69-100x10	833,4*	834,6	146,07	833,4*	<u>834,1</u>	18,23	0,00	-0,06	8,01
Daskin95-88x8	355,8	355,8	100,83	355,8	355,8	10,77	0,00	0,00	9,36
Daskin95-150x10	43919,9	43990,4	536,32	43919,9	44005,5	73,40	0,00	0,03	7,31
Gaskell67-21x5	424,9*	424,9	0,46	424,9*	424,9	0,63	0,00	0,00	0,73
Gaskell67-22x5	585,1*	585,1	0,71	585,1*	585,1	0,39	0,00	0,00	1,80
Gaskell67-29x5	512,1*	512,1	1,66	512,1*	512,1	0,92	0,00	0,00	1,81
Gaskell67-32x5A	562,2*	562,2	1,93	562,2*	562,2	0,74	0,00	0,00	2,61
Gaskell67-32x5B	504,3*	504,3	2,29	504,3*	504,3	0,69	0,00	0,00	3,32
Gaskell67-36x5	460,4*	460,4	2,86	460,4*	460,4	0,66	0,00	0,00	4,31
Min92-27x5	3062,0*	3062,0	1,34	3062,0*	3062,0	0,78	0,00	0,00	1,72
Min92-134x8	5723,6	5751,3	271,97	<u>5719,3</u>	<u>5749,9</u>	37,92	-0,08	-0,02	7,17
Perl83-12x2	204,0*	204,0	0,04	204,0*	204,0*	0,27	0,00	0,00	0,16
Perl83-55x15	1112,1*	1112,3	23,92	1112,1*	1112,3	3,57	0,00	0,00	6,70
Perl83-85x7	1622,5*	1623,7	39,58	1622,5*	1623,8	6,66	0,00	0,01	5,94
Perl83-318x4	558680,3	565035,4	4118,51	560317,3	<u>564945,4</u>	546,74	0,29	-0,02	7,53
Perl83-318x4	670804,5	678144,5	2099,07	677345,6	682883,2	540,14	0,97	0,69	3,89
Or76-117x14	12296,9	12302,5	449,12	12296,9	<u>12299,7</u>	42,99	0,00	-0,02	10,45

Na Figura 13 pode-se observar um resumo dos resultados das versões *vDistrib*, *vParallel*, *vParallel_MPI* em comparação com a versão sequencial, usando 1, 2, 4, 8 e 16 *threads*, no critério melhor solução encontrada para o primeiro banco de teste. Os valores apresentados são uma média dos valores do parâmetro “gap/MS”. Utilizando a execução de 16 *threads* como parâmetro, pela característica de possuir o melhor valor absoluto de *speedup*, a versão que obteve os melhores resultados para o primeiro banco de teste foi a versão *vDistrib* com 0,02, seguida da versão *vParallel_MPI* com 0,07 e por último a versão *vParallel* com 0,09. Uma comparação entre o desempenho das versões paralelas considerando o *speedup* pode ser visualizada na Figura 14, onde *vDistrib* mostra o melhor desempenho por não possuir comunicação.

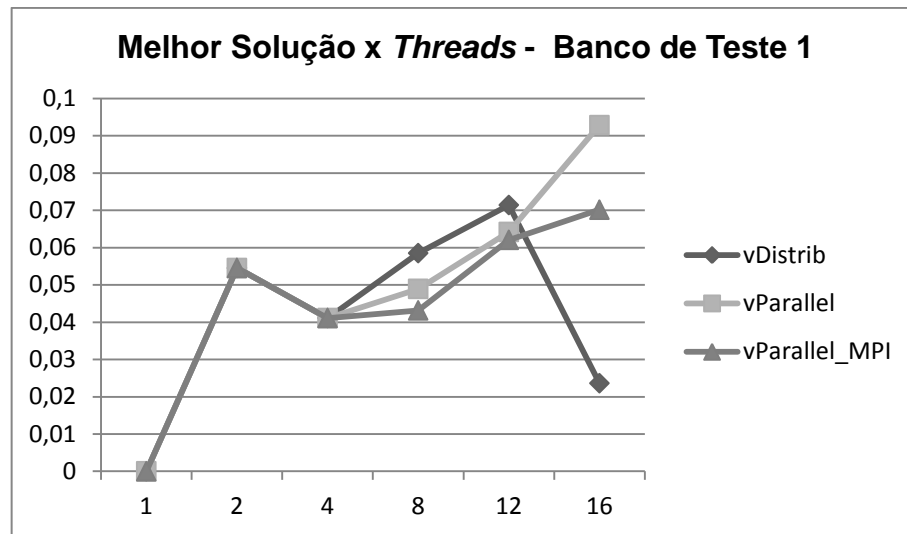


Figura 13: Melhor Solução x *Threads* – Banco de Teste 1

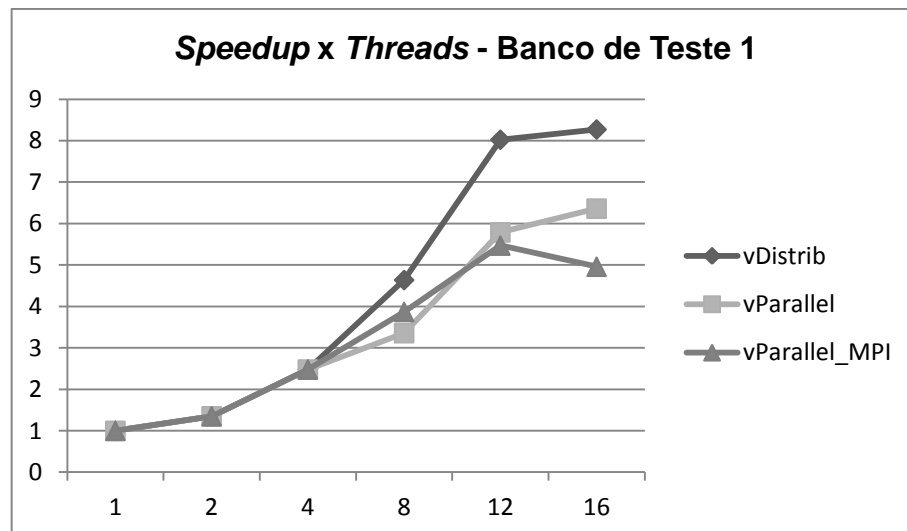


Figura 14: *Speedup* x *Threads* – Banco de Teste 1

A Tabela 12 mostra a comparação entre a versão *vDistrib*, que utiliza 16 *threads*, com versão sequencial para o segundo banco de teste. No parâmetro “gap/MS”, esta versão obteve, de 30 instâncias, 4 melhoras e 16 empates. O *speedup* médio obtido foi de 5,62 oscilando entre os valores 3,23 à 8,95. Para os parâmetros “gap/Média” e “gap/MS” a média dos valores obtidos foi respectivamente 0,11 e 0,04.

Tabela 12: Comparação do sequencial com *vDistrib* – Banco de Teste 2

Instancia	Sequencial			vDistrib – 16 Threads			gap/MSC	gap/Média	Speedup
	Custo	Média	Tempo	Custo	Média	Tempo			
20-5-1a	54793*	54793*	0,7	54793*	54793*	0,06	0,00	0,00	5,61
20-5-1b	39104*	39104*	1,1	39104*	39104*	0,07	0,00	0,00	8,48
20-5-2a	48908*	48908*	0,7	48908*	48908*	0,07	0,00	0,00	5,49
20-5-2b	37542*	37542*	0,9	37542*	37542*	0,07	0,00	0,00	7,91
50-5-1a	90111*	90111*	7,9	90111*	90111*	0,80	0,00	0,00	5,66
50-5-1b	63242*	63247,1	11,4	63242*	63247,6	0,86	0,00	0,00	7,59
50-5-2a	88298*	88298*	7,2	88298*	88298*	0,75	0,00	0,00	5,40
50-5-2b	67308*	67308*	12,1	67308*	67328,6	0,80	0,00	0,03	8,56
50-5-2BIS	84055*	84055*	13,2	84055*	84055*	0,82	0,00	0,00	8,95
50-5-2bBIS	51822*	51876,9	8,1	51822*	51876,9	0,74	0,00	0,00	6,17
50-5-3a	86203*	86203*	7,8	86203*	86203*	0,75	0,00	0,00	5,81
50-5-3b	61830*	61830*	13,1	61830*	61830*	0,87	0,00	0,00	8,65
100-5-1a	274814	275513,7	57,5	275079	<u>275418,4</u>	7,86	0,10	-0,03	4,27
100-5-1b	213896	214106,2	97,3	<u>213568</u>	<u>214091,9</u>	9,13	-0,15	-0,01	6,23
100-5-2	193671*	193789,2	58,7	193671*	193813,6	8,12	0,00	0,01	4,13
100-5-2b	157150	157192,6	95,9	<u>157129</u>	<u>157161,6</u>	8,83	-0,01	-0,02	6,27
100-5-3a	200079	200224,4	57,0	200131	200261,4	7,58	0,03	0,02	4,18
100-5-3b	152441*	152458,8	79,2	152441*	152495,6	8,05	0,00	0,02	5,66
100-10-1a	296735	308367,8	116,8	297229	<u>303528,7</u>	17,46	0,17	-1,59	3,95
100-10-1b	235611	240629,4	161,0	237828	243909,3	15,93	0,93	1,34	5,74
100-10-2a	243590	243692,6	118,0	243590	243621,8	16,23	0,00	-0,03	4,20
100-10-2b	203988	204146,1	162,2	203988	<u>204007,0</u>	15,56	0,00	-0,07	6,00
100-10-3a	252927	258384,5	119,2	253175	<u>258221,6</u>	16,09	0,10	-0,06	4,15
100-10-3b	204631	204805,8	181,7	204661	<u>204761,6</u>	16,18	0,01	-0,02	6,36
200-10-1a	475576	477085,6	1005,0	<u>475225</u>	486504,7	185,90	-0,07	1,94	3,23
200-10-1b	376698	379191,9	1605,1	<u>376472</u>	380261,4	230,64	-0,06	0,28	4,34
200-10-2a	448242	449251,4	1252,3	448600	<u>449117,5</u>	211,45	0,08	-0,03	3,41
200-10-2b	374157	374444,7	1560,3	374282	374512,4	207,98	0,03	0,02	4,49
200-10-3a	471896	481924,6	1011,3	472019	488989,8	174,99	0,03	1,44	3,29
200-10-3b	362341	363006,3	1544,3	362626	<u>362957,4</u>	209,92	0,08	-0,01	4,42

Os resultados obtidos pela comparação da versão *vParallel*, utilizando 16 *threads*, com a versão sequencial para o segundo banco de teste são mostrados na Tabela 13. Observando o parâmetro “gap/MSC”, a versão *vParallel* obteve quatro melhoras e 17 empates de 30 instâncias. O *speedup* médio foi de 4,39 oscilando de 2,66 à 6,76. Diferentemente dos resultados obtidos para o primeiro banco de teste, neste segundo banco de teste a versão *vParallel* conseguiu resultados muito bons, apresentando um valor médio no campo “gap/MSC” de 0,03 e melhorando a média em sete instâncias, obtendo o valor médio no parâmetro “gap/Média” de -0,02.

Tabela 13: Comparação do sequencial com *vParallel* – Banco de Teste 2

Instancia	Sequencial			vParallel – 16 Threads			gap/MSC	gap/Média	Speedup
	Custo	Média	Tempo	Custo	Média	Tempo			
20-5-1a	54793*	54793*	0,7	54793*	54793*	0,10	0,00	0,00	3,53
20-5-1b	39104*	39104*	1,1	39104*	39104*	0,11	0,00	0,00	5,60
20-5-2a	48908*	48908*	0,7	48908*	48908*	0,10	0,00	0,00	3,53
20-5-2b	37542*	37542*	0,9	37542*	37542*	0,10	0,00	0,00	5,21
50-5-1a	90111*	90111*	7,9	90111*	90111*	1,08	0,00	0,00	4,21
50-5-1b	63242*	63247,1	11,4	63242*	63247,1	1,06	0,00	0,00	6,13
50-5-2a	88298*	88298*	7,2	88298*	88298*	0,96	0,00	0,00	4,25
50-5-2b	67308*	67308*	12,1	67308*	67314,5	1,09	0,00	0,01	6,25
50-5-2BIS	84055*	84055*	13,2	84055*	84055*	1,10	0,00	0,00	6,71
50-5-2bBIS	51822*	51876,9	8,1	51822*	<u>51855,4</u>	1,01	0,00	-0,04	4,48
50-5-3a	86203*	86203*	7,8	86203*	86203*	1,03	0,00	0,00	4,22
50-5-3b	61830*	61830*	13,1	61830*	61830*	1,11	0,00	0,00	6,76
100-5-1a	274814	275513,7	57,5	275079	275554,2	9,53	0,10	0,01	3,53
100-5-1b	213896	214106,2	97,3	213991	214125,1	10,66	0,04	0,01	5,34
100-5-2	193671*	193789,2	58,7	193752	193920,6	10,13	0,04	0,07	3,31
100-5-2b	157150	157192,6	95,9	<u>157129</u>	<u>157168,1</u>	10,73	-0,01	-0,02	5,16
100-5-3a	200079	200224,4	57,0	200079	200227,9	9,37	0,00	0,00	3,38
100-5-3b	152441*	152458,8	79,2	152441*	152715,2	9,17	0,00	0,17	4,97
100-10-1a	296735	308367,8	116,8	<u>293539</u>	<u>304931,6</u>	20,43	-1,09	-1,13	3,38
100-10-1b	235611	240629,4	161,0	238160	240857,6	18,59	1,07	0,09	4,92
100-10-2a	243590	243692,6	118,0	243590	243809,5	19,99	0,00	0,05	3,41
100-10-2b	203988	204146,1	162,2	203988	<u>204026,0</u>	18,30	0,00	-0,06	5,10
100-10-3a	252927	258384,5	119,2	<u>252890</u>	<u>256303,7</u>	20,41	-0,01	-0,81	3,27
100-10-3b	204631	204805,8	181,7	204631	204972,8	19,52	0,00	0,08	5,27
200-10-1a	475576	477085,6	1005,0	477113	482801,5	225,28	0,32	1,18	2,66
200-10-1b	376698	379191,9	1605,1	376739	<u>377607,1</u>	266,79	0,01	-0,42	3,75
200-10-2a	448242	449251,4	1252,3	448986	449340,8	238,89	0,17	0,02	3,02
200-10-2b	374157	374444,7	1560,3	<u>373948</u>	374649,1	233,37	-0,06	0,05	4,00
200-10-3a	471896	481924,6	1011,3	472580	<u>481855,3</u>	207,20	0,14	-0,01	2,78
200-10-3b	362341	363006,3	1544,3	363050	363624,2	253,54	0,20	0,17	3,66

A comparação dos resultados obtidos entre as versões *vParallel_MPI* e a versão sequencial para o segundo banco de testes e utilizando 16 *threads* podem ser visualizados na Tabela 14. A versão *vParallel_MPI*, que utiliza o “gap/MSC” como parâmetro, de 30 instâncias, apresentou cinco melhoras e empatou em 15 resultados. O *speedup* obteve na média 4,17, oscilando entre os valores de 0,55 à 7,46. Apesar de obter cinco melhoras no parâmetro “gap/MSC”, a média foi de 0,09 e seu gap/Médio foi no geral pior que a versão sequencial, obtendo na média de seus valores 0,23.

Tabela 14: Comparação do sequencial com *vParallel_MPI* – Banco de Teste 2

Instancia	Sequencial			vParallel_MPI – 16 Threads			gap/MSC	gap/Média	Speedup
	Custo	Média	Tempo	Custo	Média	Tempo			
20-5-1a	54793*	54793*	0,7	54793*	54793*	0,10	0,00	0,00	0,60
20-5-1b	39104*	39104*	1,1	39104*	39104*	0,11	0,00	0,00	1,14
20-5-2a	48908*	48908*	0,7	48908*	48908*	0,10	0,00	0,00	0,55
20-5-2b	37542*	37542*	0,9	37542*	37542*	0,10	0,00	0,00	1,33
50-5-1a	90111*	90111*	7,9	90111*	90111*	1,08	0,00	0,00	3,24
50-5-1b	63242*	63247,1	11,4	63242*	63258,3	1,06	0,00	0,02	5,01
50-5-2a	88298*	88298*	7,2	88298*	88367,0	0,96	0,00	0,08	3,57
50-5-2b	67308*	67308*	12,1	67308*	67358,8	1,09	0,00	0,08	5,56
50-5-2BIS	84055*	84055*	13,2	84055*	84055*	1,10	0,00	0,00	7,46
50-5-2bBIS	51822*	51876,9	8,1	51883,0	51855,4	1,01	0,12	0,01	4,00
50-5-3a	86203*	86203*	7,8	86203*	86203*	1,03	0,00	0,00	3,94
50-5-3b	61830*	61830*	13,1	61830*	61830*	1,11	0,00	0,00	7,00
100-5-1a	274814	275513,7	57,5	275079	<u>275435,7</u>	9,53	0,10	-0,03	3,97
100-5-1b	213896	214106,2	97,3	213945	214124,6	10,66	0,02	0,01	6,24
100-5-2	193671*	193789,2	58,7	193671	193893,2	10,13	0,00	0,05	3,99
100-5-2b	157150	157192,6	95,9	<u>157144</u>	<u>157187,4</u>	10,73	0,00	0,00	5,95
100-5-3a	200079	200224,4	57,0	200131	200348,4	9,37	0,03	0,06	3,93
100-5-3b	152441*	152458,8	79,2	152466	152709,8	9,17	0,02	0,16	5,76
100-10-1a	296735	308367,8	116,8	302167	310058,8	20,43	1,80	0,55	3,98
100-10-1b	235611	240629,4	161,0	236495	242703,4	18,59	0,37	0,85	5,45
100-10-2a	243590	243692,6	118,0	243590	<u>243675,5</u>	19,99	0,00	-0,01	3,96
100-10-2b	203988	204146,1	162,2	203988	<u>204102,5</u>	18,30	0,00	-0,02	6,05
100-10-3a	252927	258384,5	119,2	252992	<u>257952,3</u>	20,41	0,03	-0,17	3,97
100-10-3b	204631	204805,8	181,7	<u>204567</u>	204846,0	19,52	-0,03	0,02	5,95
200-10-1a	475576	477085,6	1005,0	476979	499542,7	225,28	0,29	4,50	3,09
200-10-1b	376698	379191,9	1605,1	<u>375961</u>	379296,9	266,79	-0,20	0,03	4,26
200-10-2a	448242	449251,4	1252,3	448934	449354,4	238,89	0,15	0,02	3,39
200-10-2b	374157	374444,7	1560,3	<u>373928</u>	374455,6	233,37	-0,06	0,00	4,64
200-10-3a	471896	481924,6	1011,3	<u>471863</u>	483734,0	207,20	-0,01	0,37	3,04
200-10-3b	362341	363006,3	1544,3	<u>362253</u>	363610,9	253,54	-0,02	0,17	4,21

Na Figura 15 pode-se observar os resultados das versões *vDistrib*, *vParallel*, *vParallel_MPI* em comparação com a versão sequencial usando 1, 2, 4, 8, 12 e 16 *threads*, no critério melhor solução encontrada para o segundo banco de teste. Os valores apresentados são uma média dos valores do parâmetro “gap/MSC”. Utilizando novamente a execução de 16 *threads* como parâmetro, a versão que obteve os melhores resultados para o segundo banco de teste foi a versão *vParallel* com o valor de 0,03, seguido da versão *vDistrib* com 0,04 e por último, a versão *vParallel_MPI* com 0,09.

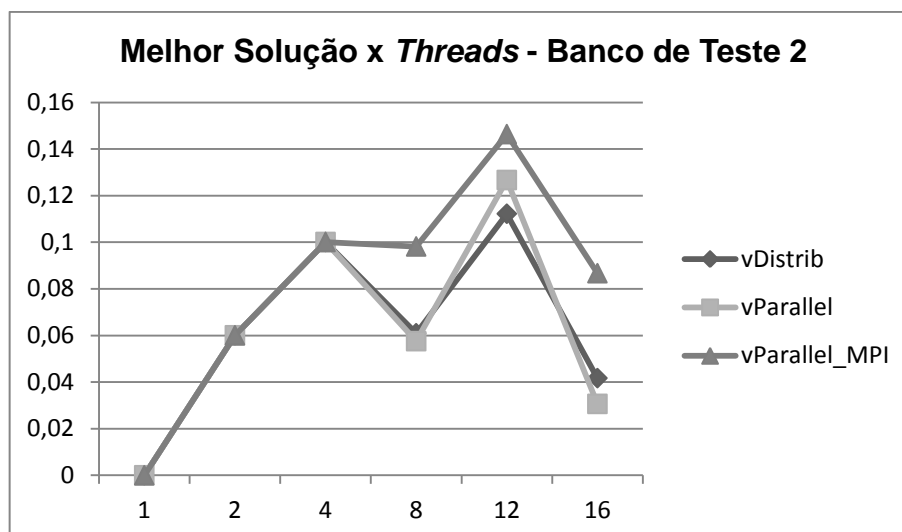


Figura 15: Melhor Solução x Threads – Banco de Teste 2

Assim como para o banco de teste 1, no critério *Speedup*, a versão *vDistrib* também alcança os melhores resultados, como pode ser observado na Figura 16.

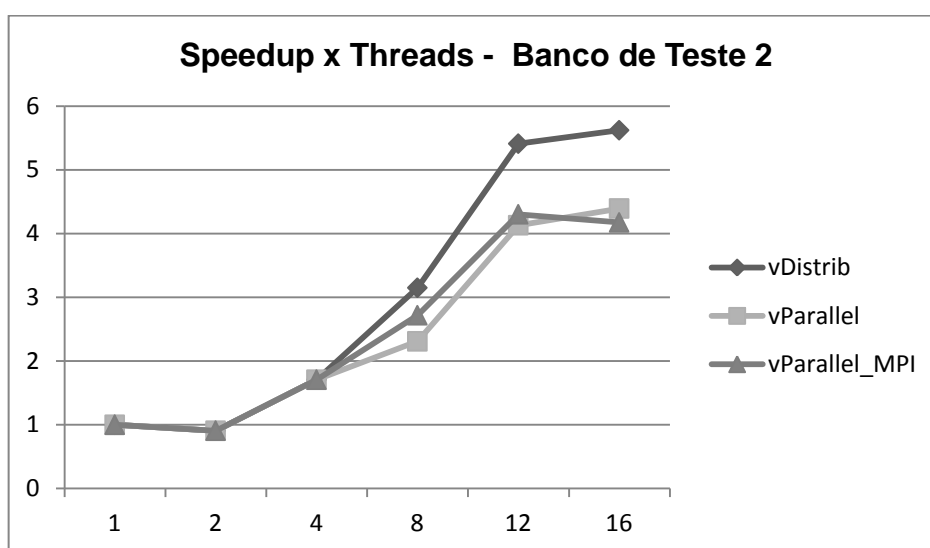


Figura 16: Speedup x Threads – Banco de Teste 2

Os resultados da comparação da versão *vDistrib*, utilizando 16 *threads*, com a versão sequencial para o terceiro banco de teste é apresentado na Tabela 15. Utilizando o “gap/MS” como parâmetro, de 36 instâncias, apresentou 15 melhoras e empatou em 18 resultados. O *speedup* obteve na média 5,42, oscilando entre os valores de 3,98 à 7,57. Os resultados foram satisfatórios por encontrar um grande número de melhores soluções que a versão sequencial e, como consequência, obteve como média do parâmetro “gap/MS” o valor de -0,05. Além disso se

mostrou mais robusto que a versão sequencial conseguindo melhorar 29 médias, apresentando como média do parâmetro “gap/Média” o valor de -0,30.

Tabela 15: Comparação da versão sequencial com *vDistrib* – Banco de Teste 3

Instancia	Sequencial			vDistrib – 16 Threads			gap/MSC	gap/Média	Speedup
	Custo	Média	Tempo	Custo	Média	Tempo			
111112	1467,68*	1469,60	166,90	1467,68	<u>1467,85</u>	28,44	0,00	-0,12	5,87
111122	1448,37	1449,11	263,12	1448,37	<u>1449,03</u>	45,38	0,00	-0,01	5,80
111212	1394,80*	1396,17	119,16	1394,80	<u>1394,80</u>	21,51	0,00	-0,10	5,54
111222	1432,29	1432,29	276,92	1432,29	1432,29	44,84	0,00	0,00	6,18
112112	1167,16*	1167,16	161,77	1167,16	1167,16	23,81	0,00	0,00	6,80
112122	1102,24	1102,24	299,58	1102,24	1102,24	39,59	0,00	0,00	7,57
112212	791,66	792,69	148,94	791,66	<u>792,20</u>	27,88	0,00	-0,06	5,34
112222	728,30	779,39	244,92	728,30	<u>728,30</u>	34,93	0,00	-7,02	7,01
113112	1238,49	1238,86	155,50	1238,49	1238,93	27,15	0,00	0,01	5,73
113122	1245,31	1245,33	356,76	1245,31	1245,33	60,88	0,00	0,00	5,86
113212	916,05	916,05	155,86	916,05	916,05	22,99	0,00	0,00	6,78
113222	1018,29	1018,29	265,34	1018,29	1018,29	52,57	0,00	0,00	5,05
131112	1907,92	1915,62	479,98	<u>1900,41</u>	<u>1913,88</u>	101,81	-0,40	-0,09	4,71
131122	1820,78	1829,53	1189,45	1822,74	<u>1828,52</u>	263,11	0,11	-0,06	4,52
131212	1964,93	1970,01	607,09	<u>1960,02</u>	<u>1969,43</u>	96,15	-0,25	-0,03	6,31
131222	1795,84	1801,85	1110,93	<u>1792,77</u>	<u>1799,16</u>	228,79	-0,17	-0,15	4,86
132112	1443,32	1443,97	733,80	1443,32	<u>1443,78</u>	105,45	0,00	-0,01	6,96
132122	1431,32	1436,01	1228,26	<u>1429,87</u>	<u>1434,90</u>	205,18	-0,10	-0,08	5,99
132212	1208,54	1209,61	581,81	1208,54	<u>1208,90</u>	117,76	0,00	-0,06	4,94
132222	927,81	937,62	1453,98	<u>927,14</u>	<u>934,79</u>	302,74	-0,07	-0,30	4,80
133112	1694,18	1698,74	553,10	1694,22	<u>1695,51</u>	112,24	0,00	-0,19	4,93
133122	1392,18	1394,84	1101,12	<u>1392,01</u>	<u>1392,67</u>	211,56	-0,01	-0,16	5,20
133212	1206,14	1206,67	647,60	1206,14	<u>1206,39</u>	115,12	0,00	-0,02	5,63
133222	1151,80	1151,94	1103,25	1151,80	<u>1151,81</u>	208,33	0,00	-0,01	5,30
121112	2239,67	2251,91	1429,79	<u>2238,59</u>	<u>2246,99</u>	310,50	-0,05	-0,22	4,60
121122	2141,43	2152,33	2467,07	<u>2137,91</u>	<u>2151,10</u>	549,85	-0,16	-0,06	4,49
121212	2204,04	2213,03	1194,71	2206,28	<u>2210,64</u>	298,59	0,10	-0,11	4,00
121222	2224,40	2230,98	2878,10	<u>2218,82</u>	<u>2227,79</u>	612,38	-0,25	-0,14	4,70
122112	2073,93	2078,63	1778,16	<u>2073,14</u>	<u>2075,85</u>	309,29	-0,04	-0,13	5,75
122122	1688,49	1697,51	3354,95	<u>1685,52</u>	<u>1691,47</u>	742,03	-0,18	-0,36	4,52
122212	1451,99	1454,37	1596,38	<u>1449,03</u>	<u>1453,45</u>	279,15	-0,20	-0,06	5,72
122222	1082,46	1084,33	2429,56	1082,46	<u>1083,44</u>	611,00	0,00	-0,08	3,98
123112	1949,25	1959,48	1644,23	<u>1947,80</u>	<u>1956,12</u>	308,93	-0,07	-0,17	5,32
123122	1910,66	1922,67	3194,95	<u>1910,08</u>	<u>1920,25</u>	700,16	-0,03	-0,13	4,56
123212	1760,13	1762,44	1420,63	1760,22	<u>1761,40</u>	247,61	0,01	-0,06	5,74
123222	1391,58	1405,92	2872,94	<u>1391,17</u>	<u>1391,93</u>	719,12	-0,03	-1,00	4,00

A Tabela 16 apresenta os resultados da comparação entre a versão *vParallel* com 16 *threads* e a versão sequencial para o terceiro banco de teste. Utilizando o “gap/MSC” como parâmetro, de 36 instâncias, foram encontrados 9 melhores resultados e 21 empates. O *speedup* obteve a média de 5,35, oscilando entre os valores de 3,92 à 6,20. O valor médio do parâmetro “gap/MSC” foi de -0,02.

No parâmetro “gap/Média” a versão *vParallel* obteve melhores resultados, conseguindo melhorar 16 médias. O valor médio deste parâmetro foi de -0,21.

Tabela 16: Comparação do sequencial com *vParallel* – Banco de Teste 3

Instancia	Sequencial			vParallel – 16 Threads			gap/MS	gap/Média	Speedup
	Custo	Média	Tempo	Custo	Média	Tempo			
111112	1467,68*	1469,60	166,90	1467,68*	<u>1469,53</u>	30,44	0,00	0,00	5,48
111122	1448,37	1449,11	263,12	1448,37	1449,11	44,84	0,00	0,00	5,87
111212	1394,80*	1396,17	119,16	1394,80*	<u>1395,50</u>	21,28	0,00	-0,05	5,60
111222	1432,29	1432,29	276,92	1432,29	1432,29	52,02	0,00	0,00	5,32
112112	1167,16*	1167,16	161,77	1167,16*	1167,16	27,55	0,00	0,00	5,87
112122	1102,24	1102,24	299,58	1102,24	1102,24	47,06	0,00	0,00	6,37
112212	791,66	792,69	148,94	791,66	<u>792,15</u>	26,53	0,00	-0,07	5,62
112222	728,30	779,39	244,92	728,30	<u>728,30</u>	43,37	0,00	-7,02	5,65
113112	1238,49	1238,86	155,50	1238,49	<u>1238,78</u>	25,97	0,00	-0,01	5,99
113122	1245,31	1245,33	356,76	1245,31	1245,43	60,80	0,00	0,01	5,87
113212	916,05	916,05	155,86	916,05	916,05	26,63	0,00	0,00	5,85
113222	1018,29	1018,29	265,34	1018,29	1018,29	48,62	0,00	0,00	5,46
131112	1907,92	1915,62	479,98	1910,16	1918,29	108,29	0,12	0,14	4,43
131122	1820,78	1829,53	1189,45	1823,79	1830,20	197,50	0,16	0,04	6,02
131212	1964,93	1970,01	607,09	1965,12	<u>1969,80</u>	103,85	0,01	-0,01	5,85
131222	1795,84	1801,85	1110,93	<u>1794,83</u>	1803,18	193,45	-0,06	0,07	5,74
132112	1443,32	1443,97	733,80	1443,32	1443,97	118,36	0,00	0,00	6,20
132122	1431,32	1436,01	1228,26	1429,75	1438,15	228,89	-0,11	0,15	5,37
132212	1208,54	1209,61	581,81	1208,54	1209,64	109,44	0,00	0,00	5,32
132222	927,81	937,62	1453,98	<u>926,01</u>	943,84	255,70	-0,19	0,66	5,69
133112	1694,18	1698,74	553,10	1694,18	<u>1695,34</u>	126,63	0,00	-0,20	4,37
133122	1392,18	1394,84	1101,12	<u>1392,01</u>	<u>1393,77</u>	219,30	-0,01	-0,08	5,02
133212	1206,14	1206,67	647,60	1206,14	<u>1206,53</u>	114,64	0,00	-0,01	5,65
133222	1151,80	1151,94	1103,25	1151,80	<u>1151,82</u>	209,31	0,00	-0,01	5,27
121112	2239,67	2251,91	1429,79	<u>2239,36</u>	<u>2246,70</u>	342,18	-0,01	-0,23	4,18
121122	2141,43	2152,33	2467,07	<u>2139,90</u>	2153,16	599,92	-0,07	0,04	4,11
121212	2204,04	2213,03	1194,71	2204,04	2215,06	293,29	0,00	0,09	4,07
121222	2224,40	2230,98	2878,10	<u>2221,12</u>	<u>2230,28</u>	618,97	-0,15	-0,03	4,65
122112	2073,93	2078,63	1778,16	<u>2072,95</u>	<u>2077,10</u>	293,18	-0,05	-0,07	6,07
122122	1688,49	1697,51	3354,95	<u>1685,52</u>	<u>1695,18</u>	653,16	-0,18	-0,14	5,14
122212	1451,99	1454,37	1596,38	<u>1449,46</u>	<u>1453,60</u>	298,19	-0,17	-0,05	5,35
122222	1082,46	1084,33	2429,56	1082,46	<u>1084,02</u>	619,85	0,00	-0,03	3,92
123112	1949,25	1959,48	1644,23	1950,02	1959,89	325,30	0,04	0,02	5,05
123122	1910,66	1922,67	3194,95	1910,66	1922,76	570,35	0,00	0,00	5,60
123212	1760,13	1762,44	1420,63	1760,58	1762,37	273,04	0,03	0,00	5,20
123222	1391,58	1405,92	2872,94	1391,58	<u>1396,33</u>	528,43	0,00	-0,69	5,44

Apresenta-se na Tabela 17 os resultados da comparação entre a versão *vParallel_MPI* com 16 *threads* e a versão sequencial para o terceiro banco de teste. Utilizando o “gap/MS” como parâmetro, de 36 instâncias, apresentou 15 melhoras e empatou em 17 resultados. O *speedup* obteve a média de 6,03, oscilando entre os valores de 5,02 à 7,56. Os resultados da versão *vParallel_MPI* foram competitivos

tanto no parâmetro “gap/MSC” quanto “gap/Média”, obtendo como média nestes parâmetros os valores -0,05 e -0,19. Para o parâmetro “gap/Média” valores melhores foram encontrados para 15 instâncias.

Tabela 17: Comparação do sequencial com *vParallel_MPI* – Banco de Teste 3

Instancia	Sequencial			vParallel_MPI – 16 Threads			gap/MSC	gap/Média	Speedup
	Custo	Média	Tempo	Custo	Média	Tempo			
111112	1467,68*	1469,60	166,90	1467,68*	1470,57	23,42	0,00	0,07	7,13
111122	1448,37	1449,11	263,12	1448,37	1449,11	48,10	0,00	0,00	5,47
111212	1394,80*	1396,17	119,16	1394,80*	1395,83	19,17	0,00	-0,02	6,22
111222	1432,29	1432,29	276,92	1432,29	1432,29	41,33	0,00	0,00	6,70
112112	1167,16*	1167,16	161,77	1167,16*	1167,59	22,92	0,00	0,04	7,06
112122	1102,24	1102,24	299,58	1102,24	1102,24	39,90	0,00	0,00	7,51
112212	791,66	792,69	148,94	791,66	<u>791,94</u>	26,07	0,00	-0,10	5,71
112222	728,30	779,39	244,92	728,30	<u>728,30</u>	37,60	0,00	-7,02	6,51
113112	1238,49	1238,86	155,50	1238,49	1238,93	20,58	0,00	0,01	7,56
113122	1245,31	1245,33	356,76	1245,31	1245,60	57,79	0,00	0,02	6,17
113212	916,05	916,05	155,86	916,05	916,05	22,43	0,00	0,00	6,95
113222	1018,29	1018,29	265,34	1018,29	1018,29	47,26	0,00	0,00	5,61
131112	1907,92	1915,62	479,98	<u>1892,17</u>	<u>1912,53</u>	86,98	-0,83	-0,16	5,52
131122	1820,78	1829,53	1189,45	<u>1820,32</u>	1832,49	192,28	-0,03	0,16	6,19
131212	1964,93	1970,01	607,09	1969,23	1974,12	91,24	0,22	0,21	6,65
131222	1795,84	1801,85	1110,93	<u>1792,77</u>	1802,54	190,03	-0,17	0,04	5,85
132112	1443,32	1443,97	733,80	<u>1442,28</u>	<u>1443,54</u>	116,49	-0,07	-0,03	6,30
132122	1431,32	1436,01	1228,26	<u>1430,50</u>	1437,01	187,00	-0,06	0,07	6,57
132212	1208,54	1209,61	581,81	1208,54	<u>1209,40</u>	101,19	0,00	-0,02	5,75
132222	927,81	937,62	1453,98	928,05	939,18	236,99	0,03	0,17	6,14
133112	1694,18	1698,74	553,10	1694,18	1703,69	99,36	0,00	0,29	5,57
133122	1392,18	1394,84	1101,12	<u>1392,01</u>	<u>1394,42</u>	209,56	-0,01	-0,03	5,25
133212	1206,14	1206,67	647,60	1206,34	1206,77	99,65	0,02	0,01	6,50
133222	1151,80	1151,94	1103,25	1151,80	<u>1151,82</u>	205,65	0,00	-0,01	5,36
121112	2239,67	2251,91	1429,79	<u>2237,71</u>	<u>2249,61</u>	264,52	-0,09	-0,10	5,41
121122	2141,43	2152,33	2467,07	<u>2139,64</u>	2154,28	466,99	-0,08	0,09	5,28
121212	2204,04	2213,03	1194,71	2204,18	2215,91	251,86	0,01	0,13	4,74
121222	2224,40	2230,98	2878,10	<u>2219,76</u>	2232,26	505,68	-0,21	0,06	5,69
122112	2073,93	2078,63	1778,16	<u>2073,23</u>	<u>2077,42</u>	273,55	-0,03	-0,06	6,50
122122	1688,49	1697,51	3354,95	<u>1685,52</u>	<u>1696,00</u>	585,06	-0,18	-0,09	5,73
122212	1451,99	1454,37	1596,38	<u>1449,62</u>	<u>1453,96</u>	307,05	-0,16	-0,03	5,20
122222	1082,46	1084,33	2429,56	1082,46	<u>1084,31</u>	484,25	0,00	0,00	5,02
123112	1949,25	1959,48	1644,23	<u>1925,72</u>	<u>1954,83</u>	302,94	-1,22	-0,24	5,43
123122	1910,66	1922,67	3194,95	<u>1910,08</u>	<u>1918,88</u>	491,07	-0,03	-0,20	6,51
123212	1760,13	1762,44	1420,63	1760,13	1762,39	241,45	0,00	0,00	5,88
123222	1391,58	1405,92	2872,94	<u>1391,17</u>	<u>1401,18</u>	556,54	-0,03	-0,34	5,16

Na Figura 17 pode-se observar os resultados das versões *vDistrib*, *vParallel*, *vParallel_MPI* em comparação com a versão sequencial, usando 1, 2, 4, 8, 12 e 16 *threads*, no critério melhor solução encontrada para o terceiro banco de teste. Utilizando a execução de 16 *threads* como parâmetro, a versão que obteve os

melhores resultados para o terceiro banco de teste foi a versão *vDistrib* com -0,050, seguida da versão *vParallel_MPI* com -0,049 e a versão *vParallel* com -0,017.

No critério *Speedup*, como pode ser observado na Figura 18, a versão que obtém o melhor desempenho é a versão *vParallel_MPI*, diferentemente do primeiro e segundo bancos de teste. Como a fase de refinamento no banco de teste 3 têm uma maior importância na qualidade da solução que o bancos de teste 1 e 2, compartilhar a melhor solução com todas as *threads* durante esta fase fez com que a versão *vParallel_MPI* conseguisse reduzir o tempo de execução desta etapa evitando que as diversas *threads* executassem um maior número de iterações procurando melhorar diferentes soluções.

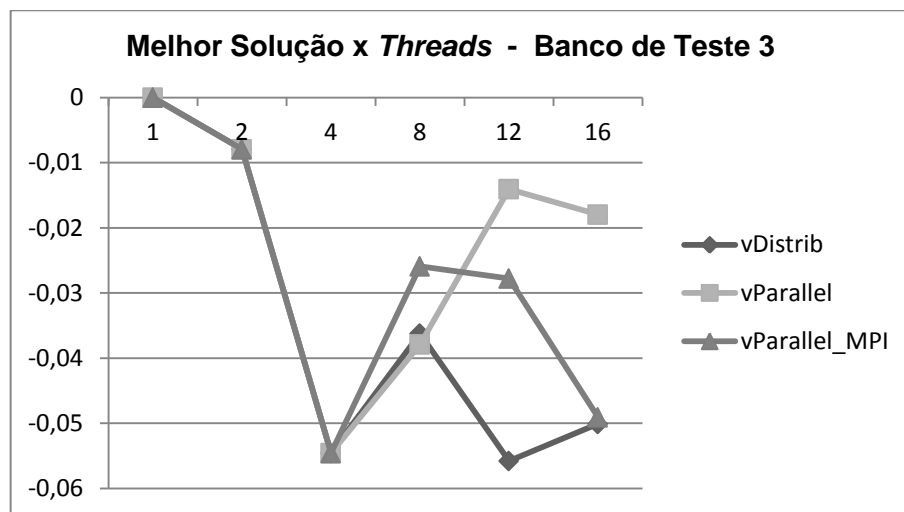


Figura 17: Melhor Solução x Threads – Banco de Teste 3

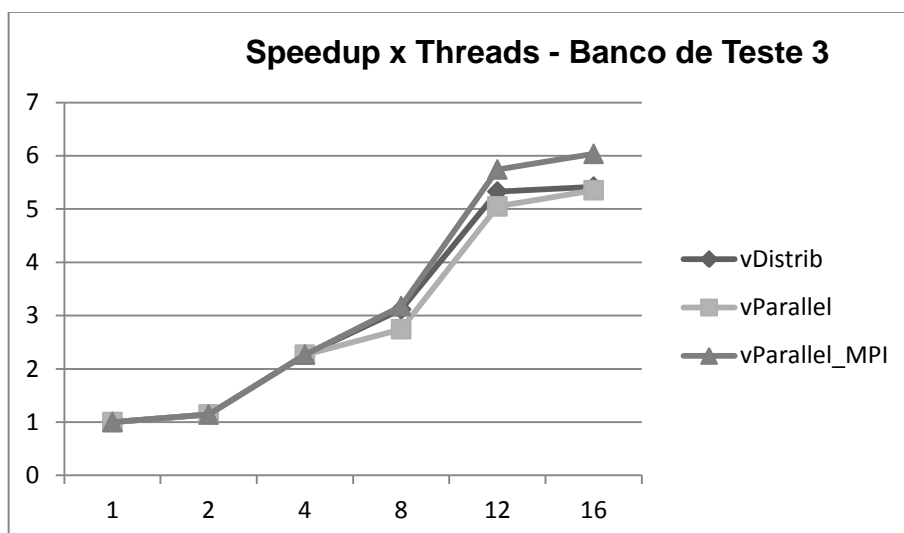


Figura 18: Speedup x Threads – Banco de Teste 3

Dado estes resultados, a etapa seguinte do trabalho consistiu em desenvolver versões que tomem proveito da comunicação e livres do paradigma da versão sequencial. Para estes testes, uma nova plataforma de execução foi disponibilizada e com isso foi possível utilizar um número maior de núcleos de processamento. A princípio a versão *vDistrib* não foi utilizada por esta não possuir comunicação e com isso não ser possível desenvolver futuramente novas versões onde a comunicação será imprescindível. É importante ressaltar que apesar da versão *vDistrib* possuir resultados que na média são ligeiramente melhores, esta melhora, no critério melhor solução, em relação a *vParallel* não ultrapassa 0,06% e 0,03% para os bancos de testes 1 e 3, respectivamente. Em relação a *vParallel_MPI* estes valores são de 0,09%, 0,04% e 0,001% para os bancos de testes 1, 2 e 3.

As versões paralelas conseguem resultados melhores para o terceiro banco de teste. Já para o primeiro e segundo bancos de teste, elas não tiram grande proveito, pois os valores apresentados pelo algoritmo sequencial já são bem competitivos. Porém, as versões paralelas conseguem resultados compatíveis em um tempo de execução bem menor, que era o objetivo inicial destes testes.

Com relação ao *speedup*, como esperado, a versão *vDistrib* apresenta na média os melhores valores, pois não existe comunicação entre as *threads*. Já a versão *vParallel* apresenta um *speedup* um pouco menor, pois gasta algum tempo com comunicação (neste caso a atualização da memória compartilhada) e a versão *vParallel_MPI* sofre um pouco mais, pois paga o preço da comunicação entre máquinas. Entretanto, como explicado anteriormente, devido a características das instâncias, a versão *vParallel_MPI* obtém o melhor *speedup* médio para o banco de teste 3.

6.2.2 Análise Complementar: Cluster OSCAR

O cluster OSCAR possui a seguinte configuração: 40 nós, cada um com dois processadores Intel Xeon E5405 Core 2 Quad 2.0 GHz, com 6MB de cache L2 e 8 MB de memória por nó. Adicionalmente, a biblioteca POSIX *Pthread* foi utilizada para programar as várias linhas de execução (*threads*). A biblioteca MPI foi utilizada para implementar a comunicação entre diferentes nós.

Nestes testes, a parametrização do campo “MAX_ITER” na versão sequencial passou a ser 32 vezes o número de depósitos disponíveis. Nas versões paralelas, este valor foi distribuído de acordo com o número de *threads*. Por exemplo, para duas *threads*, cada execução ficaria com “MAX_ITER” igual a 16 vezes o número de depósitos; oito vezes o número de depósitos para quatro *threads*; e quatro vezes para oito *threads*. Para o campo “MAX_ILS1”, a parametrização é duas vezes o número de clientes na versão sequencial e paralela, já que o número de execuções dessa região do código é sub-rotina do campo controlado por “MAX_ITER”. Na versão sequencial, o campo “MAX_ILS2” foi parametrizado com 64 vezes o número de clientes vezes o número de depósitos. Seguindo a lógica do campo “MAX_ITER”, este valor foi distribuído de acordo com o número de *threads*. Por exemplo, para duas *threads*, cada execução ficaria com “MAX_ILS2” igual a 16 vezes o número de clientes vezes o número de depósitos; oito vezes o número de clientes vezes o número de depósitos para quatro *threads*; e quatro vezes para oito *threads*.

As versões *vParallel* e *vParallel_MPI* foram executadas para verificar o seu comportamento no *cluster* OSCAR, que possui trocas de mensagens mais rápidas do que entre as máquinas utilizadas no teste anterior. Nestes testes, novamente comparam-se as melhores soluções encontradas nas versões *vParallel* e *vParallel_MPI* e a média dos resultados em 10 execuções com as melhores soluções encontradas pela versão sequencial, com a finalidade de comparar uma relação entre *speedup* e qualidade da solução. Foram realizados testes com 2, 4, 8, 16 e 32 *threads* e, vale lembrar que, as versões *vParallel* e *vParallel_MPI* são iguais para 8 *threads* ou menos, já que a troca de mensagens entre as máquinas só acontece para os testes com 16 e 32 *threads*.

Assim como no *cluster* LAPORTE, as versões paralelas *vParallel* e *vParallel_MPI* obtiveram desempenho muito parecido quando executadas no *cluster*

OSCAR. Desta forma, serão apresentados apenas os resultados da versão *vParallel_MPI* que foi escolhida por permitir a troca de mensagem entre diferentes máquinas. Assim, é possível oferecer mecanismos para que novas versões do ILS_RVND sejam configuradas em diferentes núcleos ou até mesmo em diferentes máquinas, e seja possível a troca de soluções entre elas. Os resultados são apresentados resumidamente nas Figuras 19 a 24, onde também é possível comparar o desempenho da execução da versão *vParallel_MPI* no *cluster* LAPORTE e no *cluster* OSCAR.

As Figuras 19, 20 e 21 apresentam a média do *speedup* para os Bancos de Testes 1, 2 e 3, respectivamente. De uma maneira geral, o comportamento da média foi bem próximo nos dois *clusters*. É possível observar que a medida que são acrescentados mais nós, o *speedup* ainda continua a aumentar, porém em uma taxa menor. Isto sugere que para executar em um maior número de núcleos, ainda tirando proveito do paralelismo, será necessário aumentar o número de iterações do algoritmo.

Os *speedups* obtidos atingem, em média, 1/3 do *speedup* ideal. Isso ocorre devido a natureza aleatória do algoritmo, a sobrecarga de *threads* ao adicionar a *thread* de comunicação e a grande quantidade de instâncias de fácil resolução nos bancos de teste 1 e 2. Apesar desses fatores, nas maiores instâncias, com tamanho acima de 100 nós, consegue-se obter *speedups* satisfatórios com até 2/3 do *speedup* ideal.

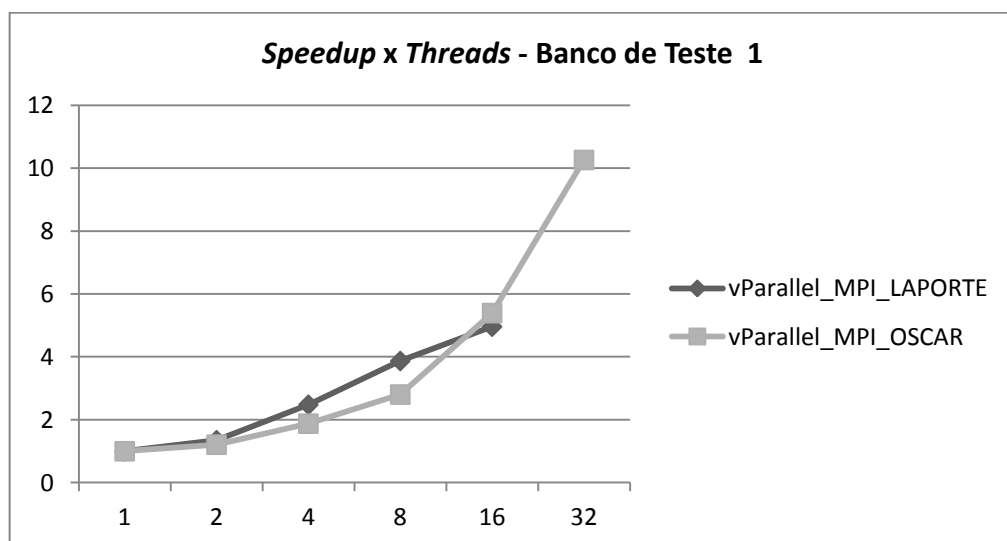


Figura 19: *Speedup x Threads* – *vParallel_MPI* – Banco de Teste 1

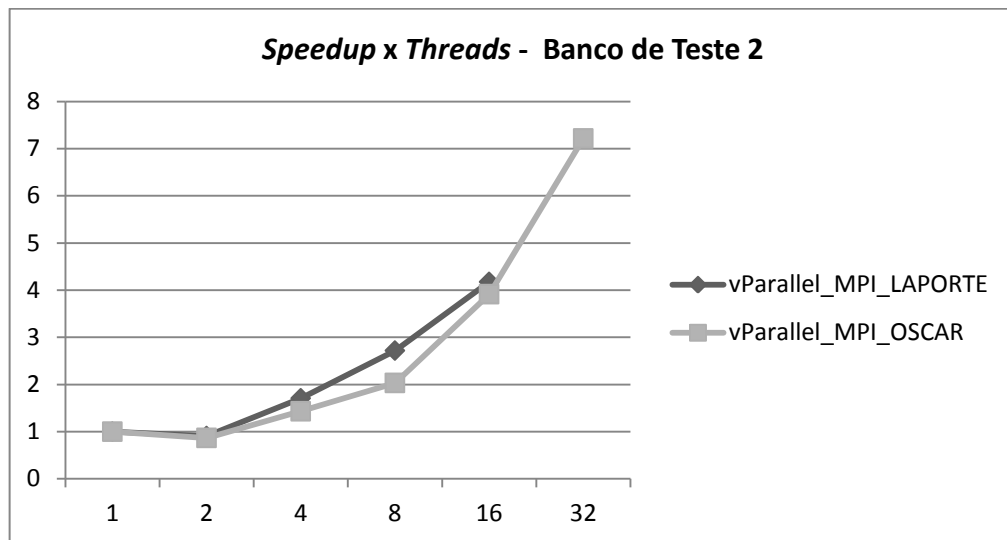


Figura 20: *Speedup x Threads* – *vParallel_MPI* – Banco de Teste 2

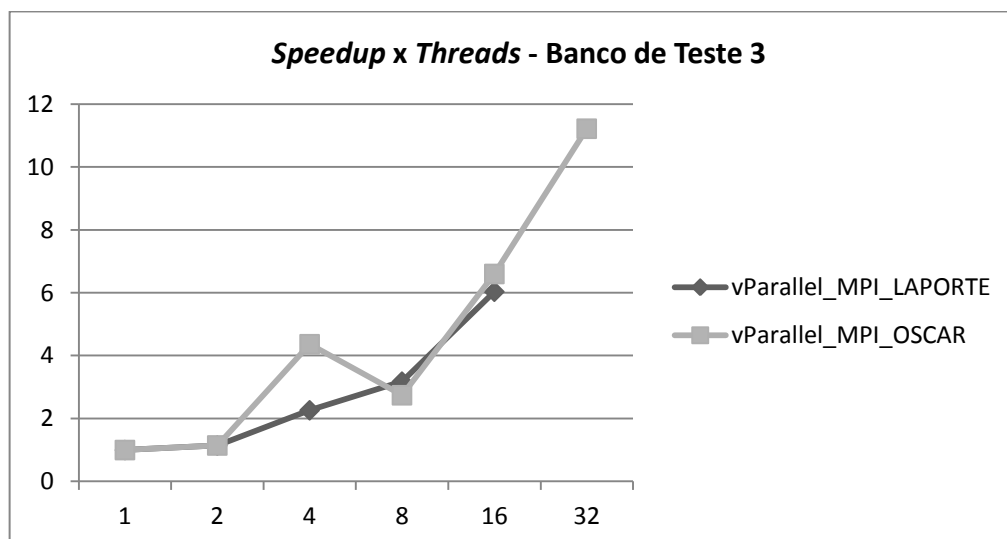


Figura 21: *Speedup x Threads* – *vParallel_MPI* – Banco de Teste 3

A seguir são apresentados os gráficos com as médias dos *gaps* das melhores soluções obtidas na execução da versão *vParallel_MPI* no *cluster* OSCAR. Na Figura 22, pode se visualizar os resultados para o Banco de Teste 1, onde a execução da versão *vParallel_MPI*, apesar de não ter produzido média de *gaps* negativos, gerou *gaps* bem mais baixos do que no *cluster* LAPORTE. Isto ocorre principalmente pelo aumento do número de iterações que foi dobrado nos testes do *cluster* OSCAR.

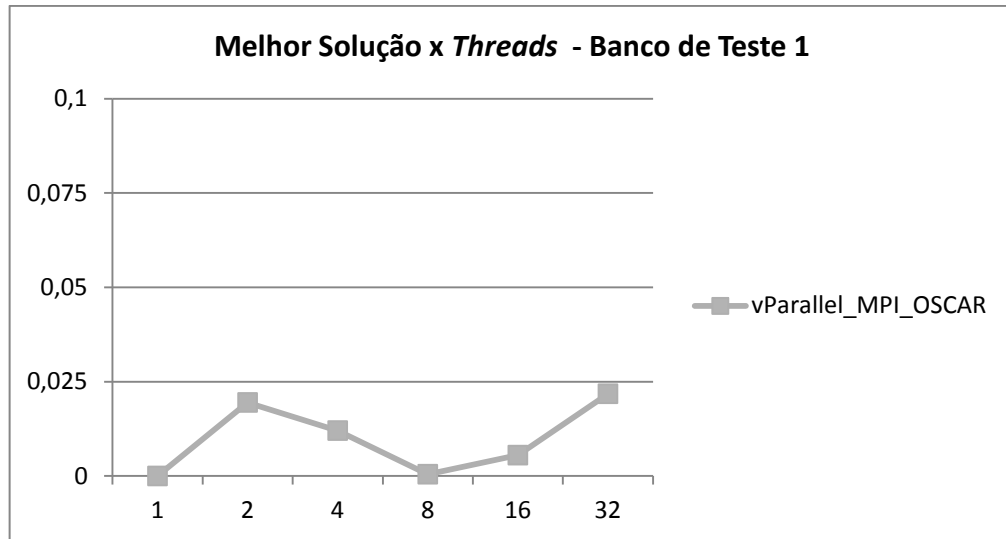


Figura 22: Melhor Solução x *Threads* – *vParallel_MPI* – Banco de Teste 1

Os resultados para o Banco de Tese 2 são apresentados na Figura 23. Como esperado, com o aumento do número iterações e consequentemente de construções, as execuções da *vParallel_MPI* no *cluster* OSCAR conseguem melhorar consideravelmente os resultados da versão sequencial. Neste caso, foram alcançadas médias de *gaps* negativos, gerando uma expressiva diferença dos resultados obtidos no *cluster* LAPORTE.

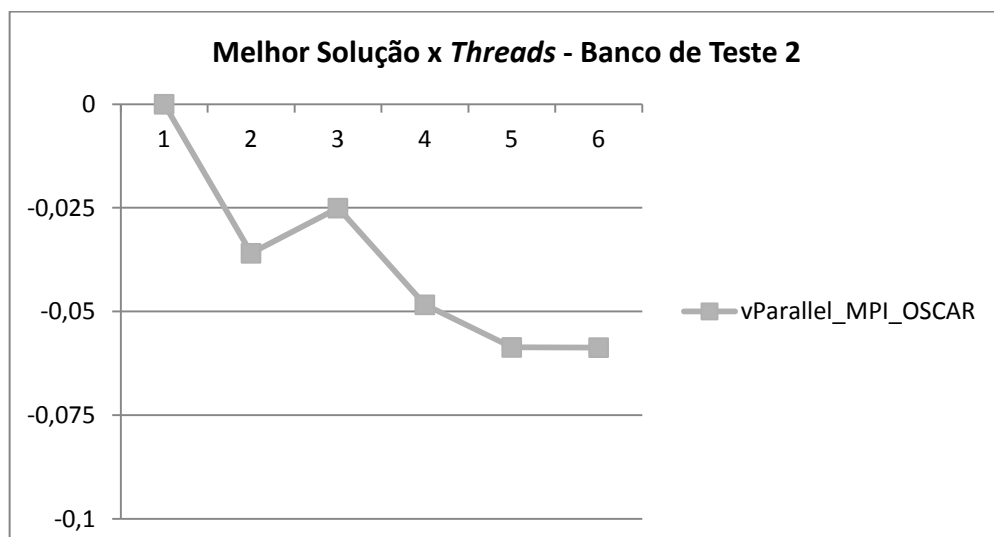


Figura 23: Melhor Solução x *Threads* – *vParallel_MPI* – Banco de Teste 2

A Figura 24 mostra os resultados para o Banco de Teste 3, onde a versão paralela obteve médias de *gaps* negativas. Porém, diferentemente dos Bancos de Testes 1 e 2, apesar do aumento do número de iterações, não se obteve melhora na

média dos *gaps* em relação ao *cluster* LAPORTE. De maneira geral, observando os valores absolutos das melhores soluções encontradas, houve uma variação bem pequena nas execuções nos dois *clusters*.

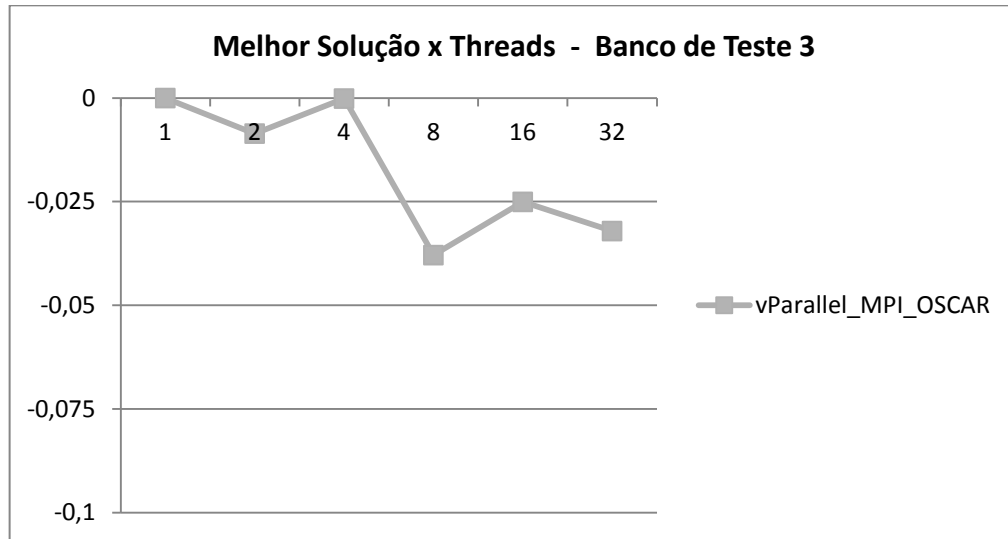


Figura 24: Melhor Solução x *Threads* – *vParallel_MPI* – Banco de Teste 3

As Tabelas 18, 19 e 20 apresentam os resultados da comparação entre a versão sequencial e a versão *vParallel_MPI*, considerando 32 *threads*, para os bancos de teste 1, 2 e 3, respectivamente. Além disso, uma outra configuração de execução para a versão *vParallel_MPI* foi testada, chamada de *vParallel_MPI_B*. A diferença é a seguinte: na versão *vParallel_MPI* existe 1 *thread* por núcleo para a execução do ILS-RVND e uma *thread* adicional, por máquina, para comunicação (totalizando 9 *threads* por máquina). Na execução de *vParallel_MPI_B* são configuradas sete *threads* para o ILS-RVND e mais uma *thread* para comunicação, considerando a máquina com 8 núcleos. Este teste teve como objetivo mostrar se é viável reservar uma *thread* para comunicação, já que desta maneira elimina-se a perda de rendimento pela sobrecarga de *threads*. Vale ressaltar que as iterações da *thread* substituída pelo comunicador não foram distribuídas para as outras *threads* na configuração usada por *vParallel_MPI_B*.

Para os resultados do banco de teste 1 (Tabela 18), considerando o parâmetro “gap/MS”, a versão *vParallel_MPI* obteve, de 19 instâncias, três índices melhores, empatando em 14 resultados. O *speedup* variou entre os valores de 0,84 a 22,18. A configuração para *vParallel_MPI_B* apresentou duas melhoras e 15 empates das 19 instâncias testadas e o *speedup* variou de 0,84 a 21,70.

Para o segundo banco de teste (Tabela 19), utilizando o “gap/MSC” como parâmetro, pode-se observar que de 30 instâncias, a versão *vParallel_MPI* obteve 11 melhoras e 17 empates e o *speedup* variou entre 2,12 e 12,56. Já para *vParallel_MPI_B* foram encontradas sete melhores soluções e 17 empates e o *speedup* foi ligeiramente melhor oscilando entre 2,23 a 12,89.

Na comparação para o terceiro banco de teste (Tabela 20), ainda observando o parâmetro “gap/MSC”, de 36 instâncias, para *vParallel_MPI* ocorreram nove melhoras e 22 empates com o *speedup* oscilando entre 7,00 e 14,71. Resultados semelhantes também foram encontrados para *vParallel_MPI_B* com nove melhoras e 22 empates. O *speedup* apresentado foi melhor, variando entre os valores de 11,00 a 18,67.

Tabela 18: Comparação da versão sequencial com as versões paralelas *vParallel_MPI* e *vParallel_MPI_B* – Banco de Teste 1

Instâncias	Sequencial			vParallel_MPI – 32 threads						vParallel_MPI_B – 32 threads					
	Custo	Média	Tempo	Custo	Média	Tempo	gap/ MSC	gap/ Média	Speedup	Custo	Média	Tempo	gap/ MSC	gap/ Média	Speedup
Christofides69-50x5	565,6*	565,6*	10,33	565,6	565,6	1,14	0,00	0,00	9,08	565,6*	566,1	1,00	0,00	0,08	10,30
Christofides69-75x10	845,4	847,4	60,97	<u>844,4</u>	<u>846,3</u>	5,25	-0,12	-0,13	11,61	844,6	847,8	5,22	-0,10	0,06	11,68
Christofides69-100x10	833,4*	833,8	245,16	833,4*	833,4*	14,59	0,00	-0,04	16,80	833,4*	833,8	15,18	0,00	0,01	16,15
Daskin95-88x8	355,8	355,8	177,08	355,8	355,8	8,93	0,00	0,00	19,83	355,8	355,8	9,13	0,00	0,00	19,40
Daskin95-150x10	43923,4	43945,1	891,92	<u>43919,9</u>	<u>43922,4</u>	59,79	-0,01	-0,05	14,92	43923,4	43954,6	61,69	0,00	0,02	14,46
Gaskell67-21x5	424,9*	424,9*	0,78	424,9*	424,9*	0,34	0,00	0,00	2,29	424,9*	424,9*	0,33	0,00	0,00	2,36
Gaskell67-22x5	585,1*	585,1*	1,26	585,1*	585,1*	0,30	0,00	0,00	4,14	585,1*	585,1*	0,28	0,00	0,00	4,54
Gaskell67-29x5	512,1*	512,1*	2,92	512,1*	512,1*	0,48	0,00	0,00	6,10	512,1*	512,1*	0,44	0,00	0,00	6,65
Gaskell67-32x5A	562,2*	562,2*	3,33	562,2*	562,2*	0,53	0,00	0,00	6,25	562,2*	562,2*	0,49	0,00	0,00	6,73
Gaskell67-32x5B	504,3*	504,3*	3,99	504,3*	504,3*	0,50	0,00	0,00	8,00	504,3*	504,3*	0,46	0,00	0,00	8,69
Gaskell67-36x5	460,4*	460,4*	4,85	460,4*	460,4*	0,55	0,00	0,00	8,87	460,4*	460,4*	0,51	0,00	0,00	9,42
Min92-27x5	3062,0*	3062,0*	2,31	3062,0*	3062,0*	0,45	0,00	0,00	5,12	3062,0*	3062,0*	0,42	0,00	0,00	5,56
Min92-134x8	5709,0	5727,9	453,86	5719,3	<u>5727,8</u>	31,58	0,18	0,00	14,37	5719,3	5731,0	33,30	0,18	0,05	13,63
Perl83-12x2	204,0*	204,0*	0,08	204,0*	204,0*	0,09	0,00	0,00	0,84	204,0*	204,0*	0,09	0,00	0,00	0,84
Perl83-55x15	1112,1*	1112,1*	39,56	1112,1*	1112,1*	4,35	0,00	0,01	9,10	1112,1*	1112,1*	3,69	0,00	0,00	10,72
Perl83-85x7	1622,5*	1622,8	64,88	1622,5*	<u>1622,7</u>	5,33	0,00	-0,01	12,17	1622,5*	1622,9	5,41	0,00	0,00	12,00
Perl83-318x4	561031,9	564058,4	6635,42	<u>560747,7</u>	<u>563958,5</u>	425,45	-0,05	-0,02	15,60	<u>558415,3</u>	<u>563553,3</u>	423,58	-0,47	-0,09	15,66
Perl83-318x4	670469,4	678657,3	3475,53	673233,8	679178,6	448,99	0,41	0,08	7,74	676289,0	680654,8	421,30	0,86	0,29	8,25
Or76-117x14	12296,9	12301,6	776,62	12296,9	<u>12296,9</u>	35,01	0,00	-0,04	22,18	12296,9	<u>12300,6</u>	35,78	0,00	-0,01	21,70

Tabela 19: Comparação da versão sequencial com as versões paralelas *vParallel_MPI* e *vParallel_MPI_B* – Banco de Teste 2

Instancia	Sequencial			vParallel_MPI – 32 threads						vParallel_MPI_B – 32 threads					
	Custo	Média	Tempo	Custo	Média	Tempo	gap/MSC	gap/Média	Speedup	Custo	Média	Tempo	gap/MSC	gap/Média	Speedup
20-5-1a	54793*	54793*	0,7	54793*	54793*	0,3	0,00	0,00	2,21	54793*	54793,0*	0,3	0,00	0,00	2,32
20-5-1b	39104*	39104*	1,1	39104*	39104*	0,3	0,00	0,00	3,52	39104*	39104,0*	0,3	0,00	0,00	3,79
20-5-2a	48908*	48908*	0,7	48908*	48908*	0,3	0,00	0,00	2,12	48908*	48908,0*	0,3	0,00	0,00	2,23
20-5-2b	37542*	37542*	0,9	37542*	37542*	0,3	0,00	0,00	3,31	37542*	37542,0*	0,3	0,00	0,00	3,48
50-5-1a	90111*	90111*	7,9	90111*	90111*	1,5	0,00	0,00	5,39	90111*	90111,0*	1,3	0,00	0,00	5,98
50-5-1b	63242*	63242*	11,4	63242*	63242*	2,1	0,00	0,00	5,50	63242*	63247,6	1,5	0,00	0,01	7,38
50-5-2a	88298*	88298*	7,2	88298*	88298*	1,4	0,00	0,00	4,94	88298*	88298,0*	1,2	0,00	0,00	6,04
50-5-2b	67308*	67308*	12,1	67308*	67308*	2,2	0,00	0,00	5,52	67308*	67308,0*	2,2	0,00	0,00	5,55
50-5-2BIS	84055**	84055*	13,2	84055*	84055*	1,3	0,00	0,00	10,43	84055*	84055,0*	1,2	0,00	0,00	11,42
50-5-2bBIS	51822*	51834,2	8,1	51822*	51852,5	1,2	0,00	0,04	6,88	51822	51834,2	1,1	0,00	0,00	7,13
50-5-3a	86203*	86203*	7,8	86203*	86203*	1,2	0,00	0,00	6,42	86203*	86203,0*	1,1	0,00	0,00	6,86
50-5-3b	61830*	61830*	13,1	61830	61830	1,3	0,00	0,00	10,41	61830*	61830,0*	1,2	0,00	0,00	11,20
100-5-1a	275457,0	275527,7	57,5	<u>275079</u>	<u>275375,2</u>	6,6	-0,14	-0,06	8,77	<u>275281</u>	<u>275524,7</u>	6,4	-0,06	0,00	8,96
100-5-1b	213568,0	214015,0	97,3	213896	214038,1	7,8	0,15	0,01	12,53	213985	214108,5	7,6	0,19	0,04	12,89
100-5-2	193671*	193855,4	58,7	193671*	<u>193805,3</u>	9,2	0,00	-0,03	6,41	193671*	<u>193762,4</u>	7,4	0,00	-0,05	7,91
100-5-2b	157144,0	157160,2	95,9	<u>157095</u>	<u>157141,5</u>	9,7	-0,03	-0,01	9,90	<u>157110</u>	157169,0	8,2	-0,02	0,01	11,72
100-5-3a	200079,0	200178,9	57,0	200079	200207,2	9,6	0,00	0,01	5,94	200079	200209,1	8,5	0,00	0,02	6,69
100-5-3b	152441*	152535,9	79,2	152441*	<u>152472,3</u>	7,3	0,00	-0,04	10,84	152441*	<u>152512,1</u>	6,8	0,00	-0,02	11,70
100-10-1a	299715,0	306654,4	116,8	<u>296573</u>	<u>302385,2</u>	21,9	-1,06	-1,41	5,33	<u>296031</u>	<u>302080,2</u>	20,0	-1,24	-1,51	5,84
100-10-1b	238336,0	243705,8	161,0	<u>236682</u>	<u>243034</u>	21,8	-0,70	-0,28	7,39	<u>236705</u>	<u>239442,1</u>	20,2	-0,69	-1,78	7,96
100-10-2a	243590,0	243590,0	118,0	243590	243703,3	18,4	0,00	0,05	6,40	243590	243741,2	16,0	0,00	0,06	7,39
100-10-2b	203988,0	204007,0	162,2	203988	204008,3	12,9	0,00	0,00	12,56	203988	204033,8	13,1	0,00	0,01	12,41
100-10-3a	252890,0	255010,8	119,2	<u>252619</u>	<u>254332,9</u>	21,4	-0,11	-0,27	5,57	252951	257945,8	17,8	0,02	1,14	6,69
100-10-3b	204661,0	204796,1	181,7	<u>204567</u>	<u>204683,9</u>	18,3	-0,05	-0,05	9,92	<u>204631</u>	<u>204743,0</u>	15,6	-0,01	-0,03	11,63
200-10-1a	476001,0	484230,7	1005,0	<u>475853</u>	<u>482036,9</u>	145,2	-0,03	-0,46	6,92	476158	485160,9	142,0	0,03	0,19	7,08
200-10-1b	376338,0	377521,8	1605,1	<u>375700</u>	<u>377014,3</u>	189,9	-0,17	-0,13	8,45	376654	<u>377314,3</u>	186,1	0,08	-0,05	8,62
200-10-2a	448890,0	449301,2	1252,3	<u>448659</u>	<u>449052</u>	160,6	-0,05	-0,06	7,80	<u>448839</u>	<u>449228,4</u>	161,2	-0,01	-0,02	7,77
200-10-2b	374078,0	374360,1	1560,3	<u>373970</u>	<u>374289,1</u>	171,9	-0,03	-0,02	9,08	<u>373998</u>	374450,1	169,8	-0,02	0,02	9,19
200-10-3a	470540,0	475354,5	1011,3	472822	478980,1	137,1	0,48	0,76	7,37	471595	<u>474623,4</u>	144,5	0,22	-0,15	7,00
200-10-3b	362555,0	363236,5	1544,3	<u>362425</u>	<u>363183,1</u>	179,0	-0,04	-0,01	8,63	362648	363545,0	167,5	0,03	0,08	9,22

Tabela 20: Comparação da versão sequencial com as versões paralelas *vParallel_MPI* e *vParallel_MPI_B* – Banco de Teste 3

Instancia	Sequencial			vParallel_MPI – 32 threads						vParallel_MPI_B – 32 threads					
	Custo	Média	Tempo	Custo	Média	Tempo	gap/MSC	gap/Média	Speedup	Custo	Média	Tempo	gap/MSC	gap/Média	Speedup
111112	1467,68*	1467,85	264,89	1467,68*	1467,68*	23,6	0,00	-0,01	11,23	1467,68*	1468,12	20,14	0,00	0,02	13,15
111122	1448,37	1448,94	455,94	1448,37	1449,03	39,4	0,00	0,01	11,57	1448,37	1449,03	33,87	0,00	0,01	13,46
111212	1394,80*	1395,83	253,43	1394,80*	<u>1394,96</u>	17,2	0,00	-0,06	14,71	1394,80*	<u>1395,48</u>	13,57	0,00	-0,02	18,67
111222	1432,29	1432,29	443,03	1432,29	1432,29	37,6	0,00	0,00	11,77	1432,29	1432,29	34,51	0,00	0,00	12,84
112112	1167,16*	1167,16*	271,16	1167,16*	1167,16*	20,2	0,00	0,00	13,45	1167,16*	1167,16*	18,12	0,00	0,00	14,96
112122	1102,24	1102,24	525,91	1102,24	1102,24	32,4	0,00	0,00	16,24	1102,24	1102,24	34,46	0,00	0,00	15,26
112212	791,66	792,23	294,79	791,66	<u>791,80</u>	21,9	0,00	-0,05	13,46	791,66	<u>792,09</u>	16,95	0,00	-0,02	17,39
112222	728,30	779,31	429,50	728,30	<u>728,30</u>	30,4	0,00	-7,01	14,15	728,30	<u>728,30</u>	30,78	0,00	-7,01	13,95
113112	1238,49	1238,56	230,88	1238,49	1238,71	26,5	0,00	0,01	8,70	1238,49	1238,64	17,75	0,00	0,01	13,01
113122	1245,31	1245,42	558,59	1245,31	<u>1245,33</u>	54,8	0,00	-0,01	10,19	1245,31	<u>1245,33</u>	40,29	0,00	-0,01	13,86
113212	916,05	916,05	256,67	916,05	916,05	18,9	0,00	0,00	13,61	916,05	916,05	17,72	0,00	0,00	14,48
113222	1018,29	1018,29	461,12	1018,29	1018,29	44,3	0,00	0,00	10,40	1018,29	1018,29	34,17	0,00	0,00	13,49
131112	1900,76	1909,55	884,56	1900,76	1910,03	80,2	0,00	0,03	11,03	<u>1898,59</u>	1913,37	63,77	-0,11	0,20	13,87
131122	1820,12	1827,69	2037,65	1820,32	<u>1824,46</u>	220,8	0,01	-0,18	9,23	1824,09	1829,57	146,07	0,22	0,10	13,95
131212	1968,05	1972,77	932,00	<u>1964,93</u>	<u>1968,81</u>	85,3	-0,16	-0,20	10,92	<u>1964,75</u>	<u>1969,25</u>	77,80	-0,17	-0,18	11,98
131222	1795,84	1802,10	1661,42	<u>1792,77</u>	<u>1798,89</u>	183,7	-0,17	-0,18	9,04	1800,43	1803,14	157,93	0,25	0,06	10,52
132112	1443,32	1444,04	1198,68	1443,32	<u>1443,61</u>	102,1	0,00	-0,03	11,73	1443,32	<u>1443,32</u>	89,86	0,00	-0,05	13,34
132122	1429,30	1434,09	2001,29	1429,42	<u>1433,76</u>	190,5	0,01	-0,02	10,51	1429,30	<u>1433,42</u>	148,34	0,00	-0,05	13,49
132212	1208,54	1209,15	1016,41	1208,54	<u>1208,81</u>	81,5	0,00	-0,03	12,47	1208,54	1209,32	75,36	0,00	0,01	13,49
132222	926,38	929,18	2808,38	<u>925,07</u>	<u>928,61</u>	215,0	-0,14	-0,06	13,06	<u>924,68</u>	<u>928,81</u>	182,21	-0,18	-0,04	15,41
133112	1694,18	1695,67	953,76	1694,18	<u>1694,55</u>	96,1	0,00	-0,07	9,93	1694,18	1695,80	80,43	0,00	0,01	11,86
133122	1392,01	1393,35	2064,10	1392,01	<u>1392,60</u>	185,5	0,00	-0,05	11,13	1392,01	<u>1392,78</u>	175,09	0,00	-0,04	11,79
133212	1206,14	1206,34	1100,30	<u>1198,08</u>	<u>1205,39</u>	104,6	-0,67	-0,08	10,52	1206,14	1206,31	76,04	0,00	0,00	14,47
133222	1151,80	1151,82	1996,39	1151,80	1151,80	168,0	0,00	0,00	11,88	1151,80	1151,82	153,25	0,00	0,00	13,03
121112	2239,67	2245,84	2686,85	<u>2237,71</u>	2247,06	242,6	-0,09	0,05	11,07	2239,67	<u>2244,23</u>	205,54	0,00	-0,07	13,07
121122	2137,74	2146,76	4056,17	2139,64	<u>2146,61</u>	472,9	0,09	-0,01	8,58	2141,16	2150,56	361,81	0,16	0,18	11,21
121212	2203,96	2212,10	2257,32	2209,13	2213,98	207,9	0,23	0,08	10,86	2204,18	2214,50	179,23	0,01	0,11	12,59
121222	2218,32	2225,43	4429,46	2218,95	2228,61	461,9	0,03	0,14	9,59	2223,32	2231,25	355,26	0,22	0,26	12,47
122112	2074,18	2077,27	2984,45	2074,15	<u>2075,09</u>	257,3	0,00	-0,11	11,60	<u>2072,37</u>	<u>2076,36</u>	236,93	-0,09	-0,04	12,60
122122	1685,78	1693,53	6663,55	<u>1685,52</u>	<u>1688,94</u>	581,2	-0,02	-0,27	11,46	<u>1685,52</u>	1694,53	437,41	-0,02	0,06	15,23
122212	1449,62	1452,69	2628,61	<u>1449,46</u>	<u>1452,33</u>	248,3	-0,01	-0,02	10,59	<u>1449,46</u>	<u>1451,25</u>	236,10	-0,01	-0,10	11,13
122222	1082,46	1083,23	4365,28	1082,46	1083,50	436,0	0,00	0,03	10,01	1082,46	<u>1083,05</u>	396,53	0,00	-0,02	11,01
123112	1953,12	1959,63	2617,94	<u>1948,37</u>	<u>1953,54</u>	245,9	-0,24	-0,31	10,65	<u>1947,40</u>	<u>1958,37</u>	219,77	-0,29	-0,06	11,91
123122	1910,08	1918,55	5549,89	1910,08	<u>1917,64</u>	563,1	0,00	-0,05	9,86	1910,08	<u>1916,85</u>	409,43	0,00	-0,09	13,56
123212	1760,45	1762,06	2377,76	1760,45	<u>1760,94</u>	204,1	0,00	-0,06	11,65	<u>1760,13</u>	<u>1761,96</u>	198,23	-0,02	-0,01	11,99
123222	1391,17	1403,60	4147,98	<u>1390,87</u>	<u>1391,46</u>	592,2	-0,02	-0,87	7,00	<u>1391,05</u>	<u>1396,27</u>	377,02	-0,01	-0,53	11,00

As Figuras 25 e 26 mostram um resumo destes resultados, onde podemos observar uma comparação entre as versões paralelas considerando a média das melhores soluções encontradas e a média dos *speedups*. Na Figura 25 pode se observar que os resultados para cada banco de teste pouco variam para cada versão paralela. Percentualmente a diferença entre as médias dos valores para “gap/MSO” é mínima, mostrando um equilíbrio entre as versões. Porém, podemos concluir que *vParallel_MPI_B* obtém resultados um pouco piores por ter um menor número de núcleos executando o algoritmo ILS_RVND. Por outro lado esta configuração produziu melhor *speedup*, como pode ser visualizado na Figura 26.

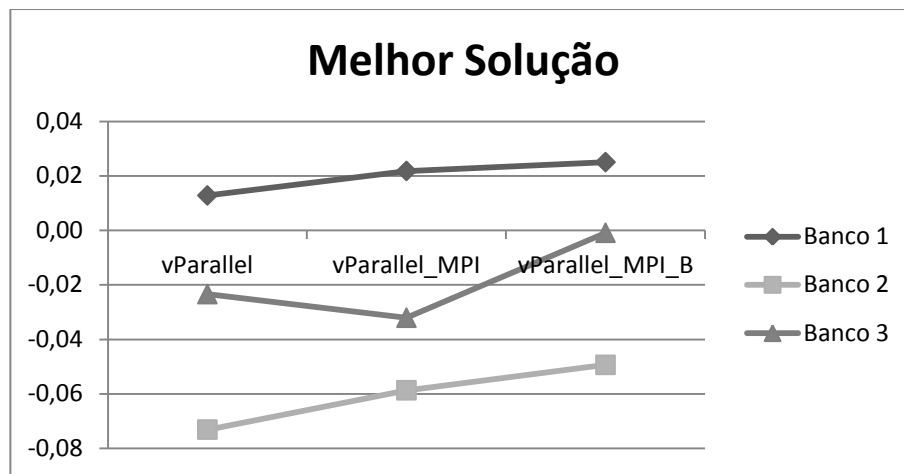


Figura 25: Gráfico Melhor Solução – Versões *vParallel*, *vParallel_MPI* e *vParallel_MPI_B* – 32 threads

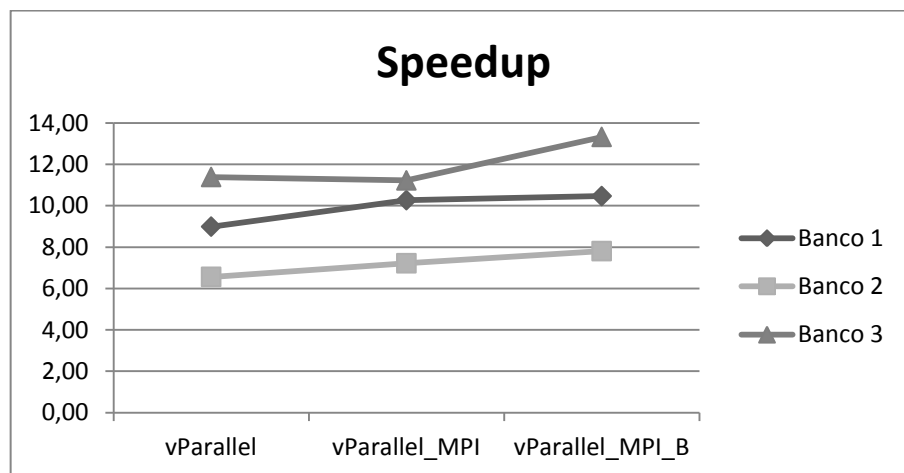


Figura 26: Gráfico *Speedup* – Versões *vParallel*, *vParallel_MPI* e *vParallel_MPI_B* – 32 threads

6.3 Versões Paralelas com Refinamento Diversificado

A versão seguinte foi baseada em testes realizados na versão sequencial onde foram observadas peculiaridades nos três bancos de testes. Foi detectado que as perturbações entre depósitos são as mais pesadas, por mexerem com várias rotas e clientes, além de não ser preciso serem realizadas tantas vezes. Após a perturbação de um depósito, é desejado que várias buscas e perturbações entre clientes sejam realizadas. Com esses dados em mãos, a probabilidade de ocorrer essas perturbações diminuíram para 50% em relação às perturbações entre clientes, gerando economia de tempo de 25% e melhorando a qualidade das soluções. Esses resultados foram obtidos através do balanceamento da melhor probabilidade de ocorrer estas perturbações nos três bancos de testes. O Banco de Teste 1 resultou em uma mesma probabilidade de ocorrer os vários tipos de perturbações; o Banco de Teste 3 sofreu uma redução de 75% na probabilidade de ocorrerem as perturbações entre depósitos; e o Banco de Teste 2 não sofreu grandes variações com a modificação das probabilidades, confirmando a importância da fase de construção neste banco de teste. Também foi detectado que a versão sequencial já encontrava o número de veículos ótimos com suas rotas relacionadas aos depósitos ótimos. Era necessário que os clientes estivessem nas rotas e posições ótimas. Para corrigir as rotas, foi inserida uma segunda fase de refinamento na versão sequencial.

A nova versão deriva da versão *vParallel_MPI* e consistiu em tirar proveito do processamento disponível, fazendo com que alguns núcleos se dedicassem a apenas perturbar e realizar buscas-locais nas melhores soluções encontradas até o momento. Um esquema gráfico pode ser visualizado na Figura 27. As quatro subversões implementadas são listadas a seguir:

- *vRefina1_CD*: um núcleo por máquina é especializado em fazer perturbações e busca locais entre depósitos e clientes;
- *vRefina2_CD*: dois núcleos por máquina são especializados em fazer perturbações e busca locais entre depósitos e clientes;
- *vRefina1_C*: um núcleo por máquina é especializado em fazer perturbações e busca locais apenas entre clientes;
- *vRefina2_C*: dois núcleos por máquina são especializados em fazer perturbações e busca locais apenas entre clientes.

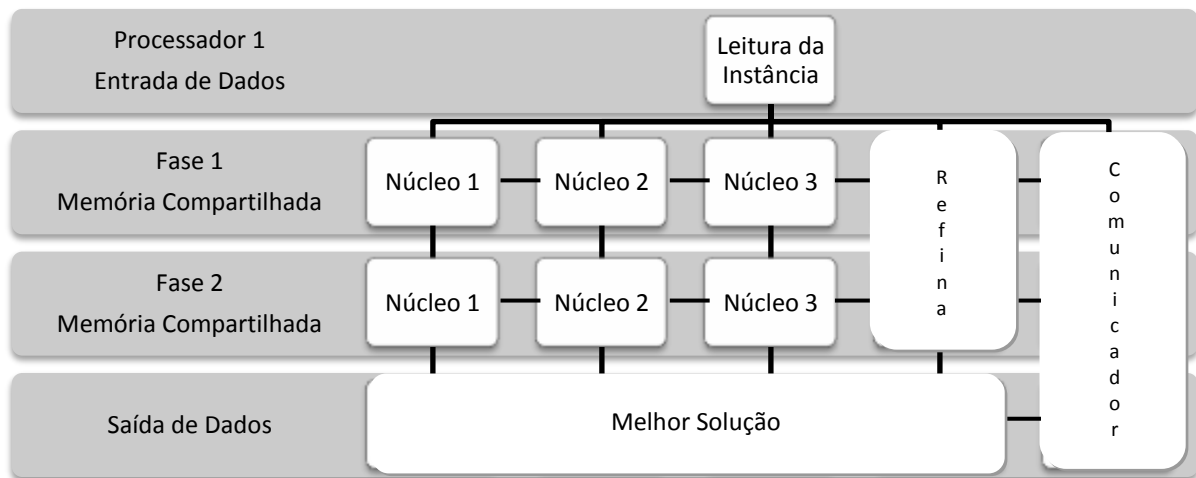


Figura 27: Esquema gráfico da versão paralela com refinamento diversificado

As Tabelas 21, 22, e 23 permitem comparar a versão sequencial com as versões paralelas *vRefina1_CD* e *vRefina2_CD* para os três bancos de testes. Para cada máquina, a versão denominada *vRefina1_CD*, reserva sete *threads* para o ILS-RVND, uma *thread* para o comunicador e uma *thread* para refinamento com movimentos entre depósitos e clientes. Na versão *vRefina2_CD* são seis *threads* reservadas para ILS-RVND, uma *thread* para o comunicador e duas *threads* para refinamento com movimentos entre depósitos e clientes. As iterações das *threads* substituídas por refinamento não foram distribuídas para as outras *threads* na versão *vRefina1_CD* e *vRefina2_CD*. O critério de parada dos refinadores é o término da última *thread* realizando ILS-RVND.

Tabela 21: Comparação da versão sequencial com as versões paralelas *vRefina1_CD* e *vRefina2_CD* – Banco de Teste 1

Instâncias	Sequencial			Paralelo <i>vRefina1_CD</i> – 32 Threads						Paralelo <i>vRefina2_CD</i> – 32 Threads					
	Custo	Média	Tempo	Custo	Média	Tempo	gap/ MSC	gap/ Média	Speedup	Custo	Média	Tempo	gap/ MSC	gap/ Média	Speedup
Christofides69-50x5	565,6*	565,6*	10,33	565,6*	565,6*	1,16	0,00	0,00	8,88	565,6*	565,6*	1,15	0,00	0,00	8,94
Christofides69-75x10	845,4	847,4	60,97	844,4	848,5	6,24	-0,12	0,14	9,77	<u>844,4</u>	847,2	6,12	-0,12	-0,02	9,97
Christofides69-100x10	833,4*	833,8	245,16	833,4*	833,9	20,15	0,00	0,02	12,17	833,4*	833,8	17,08	0,00	0,00	14,35
Daskin95-88x8	355,8	355,8	177,08	355,8	355,8	10,76	0,00	0,00	16,46	355,8	355,8	10,75	0,00	0,00	16,47
Daskin95-150x10	43923,4	43945,1	891,92	43919,9	43954,7	69,67	-0,01	0,02	12,80	<u>43919,9</u>	43969,2	66,66	-0,01	0,05	13,38
Gaskell67-21x5	424,9*	424,9*	0,78	424,9*	424,9*	0,35	0,00	0,00	2,23	424,9*	424,9*	0,37	0,00	0,00	2,11
Gaskell67-22x5	585,1*	585,1*	1,26	585,1*	585,1*	0,32	0,00	0,00	3,96	585,1*	585,1*	0,32	0,00	0,00	3,89
Gaskell67-29x5	512,1*	512,1*	2,92	512,1*	512,1*	0,50	0,00	0,00	5,83	512,1*	512,1*	0,51	0,00	0,00	5,73
Gaskell67-32x5A	562,2*	562,2*	3,33	562,2*	562,2*	0,55	0,00	0,00	6,08	562,2*	562,2*	0,56	0,00	0,00	5,89
Gaskell67-32x5B	504,3*	504,3*	3,99	504,3*	504,3*	0,51	0,00	0,00	7,81	504,3*	504,3*	0,53	0,00	0,00	7,50
Gaskell67-36x5	460,4*	460,4*	4,85	460,4*	460,4*	0,58	0,00	0,00	8,37	460,4*	460,4*	0,58	0,00	0,00	8,36
Min92-27x5	3062,0*	3062,0*	2,31	3062,0*	3062,0*	0,46	0,00	0,00	5,03	3062,0*	3062,0*	0,48	0,00	0,00	4,79
Min92-134x8	5709,0	5727,9	453,86	5719,3	5733,8	36,74	0,18	0,10	12,35	5719,3	5739,4	36,77	0,18	0,20	12,34
Perl83-12x2	204,0*	204,0*	0,08	204,0*	204,0*	0,09	0,00	0,00	0,84	204,0*	204,0*	0,09	0,00	0,00	0,84
Perl83-55x15	1112,1*	1112,1*	39,56	1112,1*	1112,1*	4,99	0,00	0,00	7,93	1112,1*	1112,1*	4,78	0,00	0,00	8,28
Perl83-85x7	1622,5*	1622,8	64,88	1622,5*	1623,0	6,53	0,00	0,01	9,93	1622,5*	1622,8	6,60	0,00	0,00	9,83
Perl83-318x4	561031,9	564058,4	6635,42	558996,3	563563,3	487,54	-0,36	-0,09	13,61	<u>560872,9</u>	<u>565780,3</u>	456,60	-0,03	0,30	14,53
Perl83-318x4	670469,4	678657,3	3475,53	674571,5	680342,4	455,48	0,61	0,25	7,63	669281,7	680229,3	461,50	-0,18	0,23	7,53
Or76-117x14	12296,9	12301,6	776,62	12296,9	12301,6	41,23	0,00	0,00	18,84	12296,9	12301,6	42,15	0,00	0,00	18,43

Tabela 22: Comparação da versão sequencial com as versões paralelas *vRefina1_CD* e *vRefina2_CD* – Banco de Teste 2

Instancia	Sequencial			Paralelo vRefina1_CD – 32 Threads						Paralelo vRefina2_CD – 32 Threads					
	Custo	Média	Tempo	Custo	Média	Tempo	gap/MSC	gap/Média	Speedup	Custo	Média	Tempo	gap/MSC	gap/Média	Speedup
20-5-1a	54793*	54793*	0,7	54793*	54793*	0,32	0,00	0,00	2,08	54793*	54793*	0,34	0,00	0,00	1,98
20-5-1b	39104*	39104*	1,1	39104*	39104*	0,31	0,00	0,00	3,47	39104*	39104*	0,37	0,00	0,00	2,89
20-5-2a	48908*	48908*	0,7	48908*	48908*	0,33	0,00	0,00	1,99	48908*	48908*	0,34	0,00	0,00	1,91
20-5-2b	37542*	37542*	0,9	37542*	37542*	0,30	0,00	0,00	3,21	37542*	37542*	0,31	0,00	0,00	3,08
50-5-1a	90111*	90111*	7,9	90111*	90111*	1,53	0,00	0,00	5,14	90111*	90111*	1,58	0,00	0,00	4,98
50-5-1b	63242*	63242*	11,4	63242*	63242	1,66	0,00	0,00	6,85	63242*	63242*	1,79	0,00	0,00	6,35
50-5-2a	88298*	88298*	7,2	88298*	88298*	1,43	0,00	0,00	4,99	88298*	88298*	1,47	0,00	0,00	4,87
50-5-2b	67308*	67308*	12,1	67308*	67308*	2,42	0,00	0,00	5,00	67308*	67314,5	2,42	0,00	0,01	5,00
50-5-2BIS	84055**	84055*	13,2	84055*	84055*	1,41	0,00	0,00	9,36	84055*	84055*	1,42	0,00	0,00	9,29
50-5-2bBIS	51822*	51834,2	8,1	51822*	51864,7	1,34	0,00	0,06	6,06	51822*	51849,3	1,31	0,00	0,03	6,18
50-5-3a	86203*	86203*	7,8	86203*	86203*	1,36	0,00	0,00	5,74	86203*	86203*	1,40	0,00	0,00	5,57
50-5-3b	61830*	61830*	13,1	61830*	61830*	1,47	0,00	0,00	8,93	61830*	61830*	1,49	0,00	0,00	8,82
100-5-1a	275457,0	275527,7	57,5	<u>275457</u>	275577,3	8,50	0,00	0,02	6,77	<u>275079</u>	<u>275506</u>	7,87	-0,14	-0,01	7,31
100-5-1b	213568,0	214015,0	97,3	213896	214076,6	10,02	0,15	0,03	9,72	213896	214113,3	9,83	0,15	0,05	9,90
100-5-2	193671*	193855,4	58,7	193671*	<u>193811,1</u>	9,49	0,00	-0,02	6,19	193671*	193864,5	10,61	0,00	0,00	5,54
100-5-2b	157144,0	157160,2	95,9	<u>157150</u>	157182,6	10,65	0,00	0,01	9,00	<u>157129</u>	157161	10,40	-0,01	0,00	9,22
100-5-3a	200079,0	200178,9	57,0	200202	200246,8	10,12	0,06	0,03	5,63	200174	200243,8	9,89	0,05	0,03	5,76
100-5-3b	152441*	152535,9	79,2	152441*	<u>152484,2</u>	8,43	0,00	-0,03	9,40	152441*	152541,9	8,63	0,00	0,00	9,18
100-10-1a	299715,0	306654,4	116,8	<u>294008</u>	<u>302043,8</u>	21,89	-1,94	-1,53	5,33	<u>297187</u>	<u>305579,1</u>	20,59	-0,85	-0,35	5,67
100-10-1b	238336,0	243705,8	161,0	<u>236873</u>	<u>239428,4</u>	22,39	-0,62	-1,79	7,19	<u>235773</u>	<u>243211,2</u>	22,16	-1,09	-0,20	7,27
100-10-2a	243590,0	243590,0	118,0	243590	243688,9	20,34	0,00	0,04	5,80	243590	243748	18,20	0,00	0,06	6,48
100-10-2b	203988,0	204007,0	162,2	203988	204045	15,59	0,00	0,02	10,41	203988	204034,2	17,04	0,00	0,01	9,52
100-10-3a	252890,0	255010,8	119,2	252890	255103,7	20,33	0,00	0,04	5,86	252890	256980,5	20,28	0,00	0,77	5,88
100-10-3b	204661,0	204796,1	181,7	<u>204567</u>	<u>204684,5</u>	18,24	-0,05	-0,05	9,96	<u>204567</u>	<u>204679,4</u>	17,79	-0,05	-0,06	10,21
200-10-1a	476001,0	484230,7	1005,0	477091	486825,6	158,66	0,23	0,53	6,33	<u>475637</u>	490167,9	160,19	-0,08	1,21	6,27
200-10-1b	376338,0	377521,8	1605,1	<u>375923</u>	<u>377301,6</u>	213,60	-0,11	-0,06	7,51	<u>376084</u>	379954,6	199,53	-0,07	0,64	8,04
200-10-2a	448890,0	449301,2	1252,3	<u>448776</u>	<u>449283,1</u>	184,38	-0,03	0,00	6,79	448933	<u>449242,4</u>	180,31	0,01	-0,01	6,95
200-10-2b	374078,0	374360,1	1560,3	<u>374150</u>	374441,2	183,02	0,02	0,02	8,53	<u>373696</u>	<u>374285,1</u>	189,46	-0,10	-0,02	8,24
200-10-3a	470540,0	475354,5	1011,3	471269	<u>478043,3</u>	155,67	0,15	0,56	6,50	472433	479136,1	153,89	0,40	0,79	6,57
200-10-3b	362555,0	363236,5	1544,3	362521	<u>363224,7</u>	190,31	-0,01	0,00	8,11	<u>362358</u>	<u>363058</u>	193,40	-0,05	-0,05	7,98

Tabela 23: Comparação da versão sequencial com as versões paralelas *vRefina1_CD* e *vRefina2_CD* – Banco de Teste 3

Instancia	Sequencial			Paralelo vRefina1_CD – 32 Threads						Paralelo vRefina2_CD – 32 Threads					
	Custo	Média	Tempo	Custo	Média	Tempo	gap/MSC	gap/Média	Speedup	Custo	Média	Tempo	gap/MSC	gap/Média	Speedup
111112	1467,68*	1467,85	264,89	1467,68*	1468,23	25,84	0,00	0,03	10,25	1467,68*	1468,12	22,31	0,00	0,02	11,87
111122	1448,37	1448,94	455,94	1448,37	1449,11	43,50	0,00	0,01	10,48	1448,37	1449,03	43,44	0,00	0,01	10,50
111212	1394,80*	1395,83	253,43	1394,80*	1395,30	18,27	0,00	-0,04	13,87	1394,80*	<u>1395,14</u>	20,29	0,00	-0,05	12,49
111222	1432,29	1432,29	443,03	1432,29	1432,29	41,17	0,00	0,00	10,76	1432,29	1432,29	37,47	0,00	0,00	11,83
112112	1167,16*	1167,16*	271,16	1167,16*	1167,16*	22,24	0,00	0,00	12,19	1167,16*	1167,16*	23,42	0,00	0,00	11,58
112122	1102,24	1102,24	525,91	1102,24	1102,24	41,08	0,00	0,00	12,80	1102,24	1102,24	40,97	0,00	0,00	12,84
112212	791,66	792,23	294,79	791,66	792,09	22,45	0,00	-0,02	13,13	791,66	<u>792,09</u>	20,73	0,00	-0,02	14,22
112222	728,30	779,31	429,50	728,30	728,30	37,69	0,00	-7,01	11,40	728,30	<u>728,30</u>	37,80	0,00	-7,01	11,36
113112	1238,49	1238,56	230,88	1238,49	1238,71	20,59	0,00	0,01	11,21	1238,49	<u>1238,49</u>	20,78	0,00	-0,01	11,11
113122	1245,31	1245,42	558,59	1245,31	1245,41	48,32	0,00	0,00	11,56	1245,31	1245,40	45,30	0,00	0,00	12,33
113212	916,05	916,05	256,67	916,05	916,05	21,07	0,00	0,00	12,18	916,05	916,05	22,37	0,00	0,00	11,47
113222	1018,29	1018,29	461,12	1018,29	1018,29	41,36	0,00	0,00	11,15	1018,29	1018,29	41,13	0,00	0,00	11,21
131112	1900,76	1909,55	884,56	1900,76	1909,65	74,70	0,00	0,00	11,84	<u>1895,83</u>	<u>1909,94</u>	82,17	-0,26	0,02	10,76
131122	1820,12	1827,69	2037,65	1820,32	1826,40	189,95	0,01	-0,07	10,73	1822,69	1829,93	156,87	0,14	0,12	12,99
131212	1968,05	1972,77	932,00	1965,12	1970,49	81,99	-0,15	-0,12	11,37	<u>1965,12</u>	<u>1968,91</u>	83,15	-0,15	-0,20	11,21
131222	1795,84	1802,10	1661,42	1792,77	1799,70	150,97	-0,17	-0,13	11,00	<u>1792,77</u>	<u>1800,63</u>	152,91	-0,17	-0,08	10,87
132112	1443,32	1444,04	1198,68	1443,32	1443,32	97,78	0,00	-0,05	12,26	1443,32	<u>1443,32</u>	99,23	0,00	-0,05	12,08
132122	1429,30	1434,09	2001,29	1429,30	1431,52	166,90	0,00	-0,18	11,99	1429,30	1437,07	160,18	0,00	0,21	12,49
132212	1208,54	1209,15	1016,41	1208,54	1208,99	83,98	0,00	-0,01	12,10	1208,54	1209,15	83,66	0,00	0,00	12,15
132222	926,38	929,18	2808,38	924,68	927,91	187,53	-0,18	-0,14	14,98	<u>924,68</u>	931,63	190,24	-0,18	0,26	14,76
133112	1694,18	1695,67	953,76	1694,18	1696,17	84,62	0,00	0,03	11,27	1694,18	1697,45	80,56	0,00	0,10	11,84
133122	1392,01	1393,35	2064,10	1392,01	1393,48	192,46	0,00	0,01	10,73	1392,01	<u>1392,64</u>	171,18	0,00	-0,05	12,06
133212	1206,14	1206,34	1100,30	1206,14	1206,31	82,51	0,00	0,00	13,33	1206,14	1206,46	82,96	0,00	0,01	13,26
133222	1151,80	1151,82	1996,39	1151,80	1151,86	188,73	0,00	0,00	10,58	1151,80	1151,82	168,04	0,00	0,00	11,88
121112	2239,67	2245,84	2686,85	2237,71	2246,73	223,63	-0,09	0,04	12,01	<u>2237,73</u>	<u>2245,71</u>	217,81	-0,09	-0,01	12,34
121122	2137,74	2146,76	4056,17	2137,45	2145,95	464,92	-0,01	-0,04	8,72	2139,64	2150,01	407,48	0,09	0,15	9,95
121212	2203,96	2212,10	2257,32	2203,53	2212,03	207,37	-0,02	0,00	10,89	<u>2203,53</u>	2212,14	216,06	-0,02	0,00	10,45
121222	2218,32	2225,43	4429,46	2214,86	2224,22	501,49	-0,16	-0,05	8,83	<u>2214,86</u>	<u>2224,43</u>	458,82	-0,16	-0,05	9,65
122112	2074,18	2077,27	2984,45	2070,43	2074,62	254,42	-0,18	-0,13	11,73	<u>2072,90</u>	<u>2075,30</u>	245,84	-0,06	-0,10	12,14
122122	1685,78	1693,53	6663,55	1685,52	1691,09	496,00	-0,02	-0,14	13,43	<u>1685,52</u>	1696,06	518,02	-0,02	0,15	12,86
122212	1449,62	1452,69	2628,61	1449,62	1453,21	224,64	0,00	0,04	11,70	<u>1449,03</u>	<u>1452,55</u>	232,05	-0,04	-0,01	11,33
122222	1082,46	1083,23	4365,28	1082,46	1083,21	466,07	0,00	0,00	9,37	1082,46	1084,65	452,85	0,00	0,13	9,64
123112	1953,12	1959,63	2617,94	1947,05	1954,87	266,53	-0,31	-0,24	9,82	<u>1946,82</u>	1959,68	240,63	-0,32	0,00	10,88
123122	1910,08	1918,55	5549,89	1910,08	1918,07	465,51	0,00	-0,02	11,92	1910,08	1922,58	466,44	0,00	0,21	11,90
123212	1760,45	1762,06	2377,76	1760,69	1762,17	216,62	0,01	0,01	10,98	<u>1760,22</u>	<u>1761,69</u>	205,05	-0,01	-0,02	11,60
123222	1391,17	1403,60	4147,98	1391,17	1391,63	429,71	0,00	-0,86	9,65	<u>1391,04</u>	<u>1391,74</u>	481,75	-0,01	-0,85	8,61

Analisando a Tabela 21, que compara as versões *vRefina1_CD* e *vRefina2_CD* com a versão sequencial para o primeiro banco de teste, verifica-se que, utilizando o “gap/MSD” como parâmetro, a versão *vRefina1_CD* obteve, de 19 instâncias, três melhoras e 14 empates com *speedup* médio de 8,97, com valores de 0,84 a 18,84. A versão *vRefina2_CD* obteve, de 19 instâncias, quatro melhoras e 14 empates, com o *speedup* oscilando entre os valores de 0,84 à 18,43 e média de 9,11.

Os resultados para o segundo banco de teste são apresentados na Tabela 22. Em relação ao parâmetro “gap/MSD”, a versão *vRefina1_CD* em 30 instâncias, teve seis melhoras e 19 empates, com o *speedup* oscilando entre os valores de 1,99 a 10,41 com média 6,60. Já a versão *vRefina2_CD* alcançou nove melhoras e 17 empates, com média do *speedup* de 6,56, variando entre os valores de 1,98 a 10,21.

Na Tabela 23 são mostrados os valores obtidos nos testes realizados para comparação da versão sequencial com as versões *vRefina1_CD* e *vRefina2_CD* para o Banco de Teste 3. Das 36 instâncias, considerando o “gap/MSD”, a versão *vRefina1_CD* apresentou 10 melhoras e 24 empates enquanto a versão *vRefina2_CD* obteve 13 melhoras 21 empates. No parâmetro “Speedup” a versão *vRefina1_CD* alcançou média 11,45 os oscilando entre os valores de 9,37 à 14,98 e a versão *vRefina2_CD* obteve na média 11,68 oscilando entre os valores de 8,61 a 14,76.

Nas Tabelas 24, 25 e 26 são apresentados os resultados obtidos na comparação entre a versão sequencial e as versões paralelas *vRefina1_C* e *vRefina2_C* para os três bancos de testes. A versão chamada *vRefina1_C*, é composta por nove *threads* por máquina. Sete destas *threads* realizam ILS-RVND, uma *thread* para o comunicador e uma *thread* refinando com movimentos apenas entre clientes. Na versão *vRefina2_C* também é composta por nove *threads*, sendo seis *threads* realizando ILS-RVND, uma *thread* para o comunicador e duas *threads* refinando com movimentos entre clientes. As iterações das *threads* substituídas por refinamento não foram distribuídas para as outras *threads* na versão *vRefina1_C* e *vRefina2_C*. O critério de parada dos refinadores é o termino da última *thread* realizando ILS-RVND.

Tabela 24: Comparação da versão sequencial com as versões paralelas *vRefina1_C* e *vRefina2_C* – Banco de Teste 1

Instâncias	Sequencial			Paralelo <i>vRefina1_C</i> – 32 Threads						Paralelo <i>vRefina2_C</i> – 32 Threads					
	Custo	Média	Tempo	Custo	Média	Tempo	gap/ MSC	gap/ Média	Speedup	Custo	Média	Tempo	gap/ MSC	gap/ Média	Speedup
Christofides69-50x5	565,6*	565,6*	10,33	565,6	565,6	1,14	0,00	0,00	9,08	565,6*	566,1	1,00	0,00	0,08	10,30
Christofides69-75x10	845,4	847,4	60,97	<u>844,4</u>	<u>846,3</u>	5,25	-0,12	-0,13	11,61	844,6	847,8	5,22	-0,10	0,06	11,68
Christofides69-100x10	833,4*	833,8	245,16	833,4*	833,4*	14,59	0,00	-0,04	16,80	833,4*	833,8	15,18	0,00	0,01	16,15
Daskin95-88x8	355,8	355,8	177,08	355,8	355,8	8,93	0,00	0,00	19,83	355,8	355,8	9,13	0,00	0,00	19,40
Daskin95-150x10	43923,4	43945,1	891,92	<u>43919,9</u>	<u>43922,4</u>	59,79	-0,01	-0,05	14,92	43923,4	43954,6	61,69	0,00	0,02	14,46
Gaskell67-21x5	424,9*	424,9*	0,78	424,9*	424,9*	0,34	0,00	0,00	2,29	424,9*	424,9*	0,33	0,00	0,00	2,36
Gaskell67-22x5	585,1*	585,1*	1,26	585,1*	585,1*	0,30	0,00	0,00	4,14	585,1*	585,1*	0,28	0,00	0,00	4,54
Gaskell67-29x5	512,1*	512,1*	2,92	512,1*	512,1*	0,48	0,00	0,00	6,10	512,1*	512,1*	0,44	0,00	0,00	6,65
Gaskell67-32x5A	562,2*	562,2*	3,33	562,2*	562,2*	0,53	0,00	0,00	6,25	562,2*	562,2*	0,49	0,00	0,00	6,73
Gaskell67-32x5B	504,3*	504,3*	3,99	504,3*	504,3*	0,50	0,00	0,00	8,00	504,3*	504,3*	0,46	0,00	0,00	8,69
Gaskell67-36x5	460,4*	460,4*	4,85	460,4*	460,4*	0,55	0,00	0,00	8,87	460,4*	460,4*	0,51	0,00	0,00	9,42
Min92-27x5	3062,0*	3062,0*	2,31	3062,0*	3062,0*	0,45	0,00	0,00	5,12	3062,0*	3062,0*	0,42	0,00	0,00	5,56
Min92-134x8	5709,0	5727,9	453,86	5719,3	<u>5727,8</u>	31,58	0,18	0,00	14,37	5719,3	5731,0	33,30	0,18	0,05	13,63
Perl83-12x2	204,0*	204,0*	0,08	204,0*	204,0*	0,09	0,00	0,00	0,84	204,0*	204,0*	0,09	0,00	0,00	0,84
Perl83-55x15	1112,1*	1112,1*	39,56	1112,1*	1112,1*	4,35	0,00	0,01	9,10	1112,1*	1112,1*	3,69	0,00	0,00	10,72
Perl83-85x7	1622,5*	1622,8	64,88	1622,5*	<u>1622,7</u>	5,33	0,00	-0,01	12,17	1622,5*	1622,9	5,41	0,00	0,00	12,00
Perl83-318x4	561031,9	564058,4	6635,42	<u>560747,7</u>	<u>563958,5</u>	425,45	-0,05	-0,02	15,60	<u>558415,3</u>	<u>563553,3</u>	423,58	-0,47	-0,09	15,66
Perl83-318x4	670469,4	678657,3	3475,53	673233,8	679178,6	448,99	0,41	0,08	7,74	676289,0	680654,8	421,30	0,86	0,29	8,25
Or76-117x14	12296,9	12301,6	776,62	12296,9	<u>12296,9</u>	35,01	0,00	-0,04	22,18	12296,9	<u>12300,6</u>	35,78	0,00	-0,01	21,70

Tabela 25: Comparação da versão sequencial com as versões paralelas *vRefina1_C* e *vRefina2_C* – Banco de Teste 2

Instancia	Sequencial			Paralelo vRefina1_C – 32 Threads						Paralelo vRefina2_C – 32 Threads					
	Custo	Média	Tempo	Custo	Média	Tempo	gap/MSC	gap/Média	Speedup	Custo	Média	Tempo	gap/MSC	gap/Média	Speedup
20-5-1a	54793*	54793*	0,7	54793*	54793*	0,32	0,00	0,00	2,09	54793*	54793*	0,34	0,00	0,00	1,99
20-5-1b	39104*	39104*	1,1	39104*	39104*	0,31	0,00	0,00	3,44	39104*	39104*	0,34	0,00	0,00	3,19
20-5-2a	48908*	48908*	0,7	48908*	48908*	0,32	0,00	0,00	2,02	48908*	48908*	0,35	0,00	0,00	1,89
20-5-2b	37542*	37542*	0,9	37542*	37542*	0,30	0,00	0,00	3,20	37542*	37542*	0,31	0,00	0,00	3,04
50-5-1a	90111*	90111*	7,9	90111*	90111*	1,59	0,00	0,00	4,95	90111*	90111*	1,66	0,00	0,00	4,75
50-5-1b	63242*	63242*	11,4	63242*	63242*	1,76	0,00	0,00	6,46	63242*	63242*	1,84	0,00	0,00	6,21
50-5-2a	88298*	88298*	7,2	88298*	88298*	1,46	0,00	0,00	4,91	88298*	88298*	1,56	0,00	0,00	4,59
50-5-2b	67308*	67308*	12,1	67308*	67308*	2,53	0,00	0,00	4,78	67308*	67314,5	2,46	0,00	0,01	4,91
50-5-2BIS	84055*	84055*	13,2	84055*	84055*	1,46	0,00	0,00	9,03	84055*	84055*	1,48	0,00	0,00	8,91
50-5-2bBIS	51822*	51834,2	8,1	51822*	51846,4	1,40	0,00	0,02	5,79	51822*	51861,5	1,46	0,00	0,05	5,56
50-5-3a	86203*	86203*	7,8	86203*	86203*	1,39	0,00	0,00	5,58	86203*	86203*	1,40	0,00	0,00	5,55
50-5-3b	61830*	61830*	13,1	61830*	61830*	1,43	0,00	0,00	9,22	61830*	61830*	1,49	0,00	0,00	8,80
100-5-1a	275457,0	275527,7	57,5	<u>275079</u>	<u>275448,7</u>	8,59	-0,14	-0,03	6,70	<u>275079</u>	<u>275414,3</u>	8,52	-0,14	-0,04	6,75
100-5-1b	213568,0	214015,0	97,3	213568	214045,1	10,26	0,00	0,01	9,48	213772	214030,7	9,76	0,10	0,01	9,97
100-5-2	193671*	193855,4	58,7	193671*	<u>193823,3</u>	11,23	0,00	-0,02	5,23	193671*	<u>193795,5</u>	10,47	0,00	-0,03	5,61
100-5-2b	157144,0	157160,2	95,9	<u>157129</u>	157167,9	10,50	-0,01	0,00	9,13	157144	157166	11,59	0,00	0,00	8,27
100-5-3a	200079,0	200178,9	57,0	200079	200196,4	10,31	0,00	0,01	5,53	200079	200207,7	11,10	0,00	0,01	5,14
100-5-3b	152441*	152535,9	79,2	152441*	<u>152451,1</u>	9,16	0,00	-0,06	8,64	152441*	152451,2	8,28	0,00	-0,06	9,57
100-10-1a	299715,0	306654,4	116,8	<u>297573</u>	<u>301502,2</u>	22,58	-0,72	-1,71	5,17	<u>297759</u>	<u>301753</u>	25,71	-0,66	-1,62	4,54
100-10-1b	238336,0	243705,8	161,0	<u>235682</u>	<u>241971,7</u>	23,44	-1,13	-0,72	6,87	<u>234531</u>	<u>238588,2</u>	23,37	-1,62	-2,14	6,89
100-10-2a	243590,0	243590,0	118,0	243590	243825	19,40	0,00	0,10	6,08	243590	243849,5	20,02	0,00	0,11	5,89
100-10-2b	203988,0	204007,0	162,2	203988	204054,5	15,96	0,00	0,02	10,16	203988	<u>203988</u>	15,61	0,00	-0,01	10,39
100-10-3a	252890,0	255010,8	119,2	<u>252590</u>	255159,4	21,12	-0,12	0,06	5,64	252890	<u>254958,3</u>	23,43	0,00	-0,02	5,09
100-10-3b	204661,0	204796,1	181,7	<u>204567</u>	<u>204696,2</u>	19,54	-0,05	-0,05	9,30	204567	<u>204689</u>	18,17	-0,05	-0,05	10,00
200-10-1a	476001,0	484230,7	1005,0	476818	486071,3	161,15	0,17	0,38	6,24	476039	<u>478862</u>	164,37	0,01	-1,12	6,11
200-10-1b	376338,0	377521,8	1605,1	<u>376077</u>	<u>377287</u>	211,63	-0,07	-0,06	7,58	<u>375868</u>	378199,8	198,45	-0,13	0,18	8,09
200-10-2a	448890,0	449301,2	1252,3	<u>448540</u>	<u>449178,9</u>	177,99	-0,08	-0,03	7,04	<u>448554</u>	<u>449168,7</u>	179,68	-0,07	-0,03	6,97
200-10-2b	374078,0	374360,1	1560,3	374227	374458	185,33	0,04	0,03	8,42	<u>373771</u>	374350,1	189,76	-0,08	0,00	8,22
200-10-3a	470540,0	475354,5	1011,3	471845	478483,2	151,08	0,28	0,65	6,69	471652	477037,1	155,93	0,24	0,35	6,49
200-10-3b	362555,0	363236,5	1544,3	<u>362452</u>	<u>363078,6</u>	186,89	-0,03	-0,04	8,26	362697	<u>363180,6</u>	185,46	0,04	-0,02	8,33

Tabela 26: Comparação da versão sequencial com as versões paralelas *vRefina1_C* e *vRefina2_C* – Banco de Teste 3

Instancia	Sequencial			Paralelo vRefina1_C – 32 Threads						Paralelo vRefina2_C – 32 Threads					
	Custo	Média	Tempo	Custo	Média	Tempo	gap/MSC	gap/Média	Speedup	Custo	Média	Tempo	gap/MSC	gap/Média	Speedup
111112	1467,68*	1467,85	264,89	1467,68*	<u>1468,12</u>	22,31	0,00	0,02	11,87	1467,68*	1468,56	22,56	0,00	0,05	11,74
111122	1448,37	1448,94	455,94	1448,37	1449,03	43,44	0,00	0,01	10,50	1448,37	1449,03	40,43	0,00	0,01	11,28
111212	1394,80*	1395,83	253,43	1394,80*	<u>1395,14</u>	20,29	0,00	-0,05	12,49	1394,80*	<u>1395,50</u>	18,90	0,00	-0,02	13,41
111222	1432,29	1432,29	443,03	1432,29	1432,29	37,47	0,00	0,00	11,83	1432,29	1432,29	39,54	0,00	0,00	11,20
112112	1167,16*	1167,16*	271,16	1167,16*	1167,16*	23,42	0,00	0,00	11,58	1167,16*	1167,16*	22,33	0,00	0,00	12,14
112122	1102,24	1102,24	525,91	1102,24	1102,24	40,97	0,00	0,00	12,84	1102,24	1102,24	40,97	0,00	0,00	12,84
112212	791,66	792,23	294,79	791,66	792,09	20,73	0,00	-0,02	14,22	791,66	792,31	22,07	0,00	0,01	13,36
112222	728,30	779,31	429,50	728,30	<u>728,30</u>	37,80	0,00	-7,01	11,36	728,30	<u>728,30</u>	37,15	0,00	-7,01	11,56
113112	1238,49	1238,56	230,88	1238,49	<u>1238,49</u>	20,78	0,00	-0,01	11,11	1238,49	1238,56	24,55	0,00	0,00	9,40
113122	1245,31	1245,42	558,59	1245,31	1245,40	45,30	0,00	0,00	12,33	1245,31	<u>1245,41</u>	48,30	0,00	0,00	11,56
113212	916,05	916,05	256,67	916,05	916,05	22,37	0,00	0,00	11,47	916,05	916,05	21,91	0,00	0,00	11,72
113222	1018,29	1018,29	461,12	1018,29	1018,29	41,13	0,00	0,00	11,21	1018,29	1018,29	42,46	0,00	0,00	10,86
131112	1900,76	1909,55	884,56	<u>1895,83</u>	1909,94	82,17	-0,26	0,02	10,76	1906,67	1916,17	76,64	0,31	0,35	11,54
131122	1820,12	1827,69	2037,65	<u>1822,69</u>	1829,93	156,87	0,14	0,12	12,99	1824,09	1830,67	151,65	0,22	0,16	13,44
131212	1968,05	1972,77	932,00	<u>1965,12</u>	<u>1968,91</u>	83,15	-0,15	-0,20	11,21	<u>1960,02</u>	<u>1969,27</u>	82,87	-0,41	-0,18	11,25
131222	1795,84	1802,10	1661,42	<u>1792,77</u>	<u>1800,63</u>	152,91	-0,17	-0,08	10,87	1801,39	1802,90	163,52	0,31	0,04	10,16
132112	1443,32	1444,04	1198,68	1443,32	<u>1443,32</u>	99,23	0,00	-0,05	12,08	1443,32	<u>1443,32</u>	96,85	0,00	-0,05	12,38
132122	1429,30	1434,09	2001,29	1429,30	1437,07	160,18	0,00	0,21	12,49	1431,20	1434,97	173,21	0,13	0,06	11,55
132212	1208,54	1209,15	1016,41	1208,54	1209,15	83,66	0,00	0,00	12,15	1208,54	1209,21	83,51	0,00	0,01	12,17
132222	926,38	929,18	2808,38	<u>924,68</u>	931,63	190,24	-0,18	0,26	14,76	<u>924,68</u>	931,82	190,54	-0,18	0,28	14,74
133112	1694,18	1695,67	953,76	1694,18	1697,45	80,56	0,00	0,10	11,84	1694,18	<u>1694,55</u>	90,39	0,00	-0,07	10,55
133122	1392,01	1393,35	2064,10	1392,01	<u>1392,64</u>	171,18	0,00	-0,05	12,06	1392,01	1393,42	178,26	0,00	0,01	11,58
133212	1206,14	1206,34	1100,30	1206,14	1206,46	82,96	0,00	0,01	13,26	1206,14	1206,35	86,85	0,00	0,00	12,67
133222	1151,80	1151,82	1996,39	1151,80	1151,82	168,04	0,00	0,00	11,88	1151,80	<u>1151,81</u>	172,03	0,00	0,00	11,60
121112	2239,67	2245,84	2686,85	<u>2237,73</u>	<u>2245,71</u>	217,81	-0,09	-0,01	12,34	<u>2237,71</u>	2247,46	238,02	-0,09	0,07	11,29
121122	2137,74	2146,76	4056,17	2139,64	2150,01	407,48	0,09	0,15	9,95	2137,94	2149,92	419,64	0,01	0,15	9,67
121212	2203,96	2212,10	2257,32	<u>2203,53</u>	2212,14	216,06	-0,02	0,00	10,45	<u>2203,75</u>	<u>2211,42</u>	245,96	-0,01	-0,03	9,18
121222	2218,32	2225,43	4429,46	<u>2214,86</u>	2224,43	458,82	-0,16	-0,05	9,65	2220,96	2227,48	409,77	0,12	0,09	10,81
122112	2074,18	2077,27	2984,45	<u>2072,90</u>	2075,30	245,84	-0,06	-0,10	12,14	<u>2070,43</u>	<u>2073,76</u>	234,10	-0,18	-0,17	12,75
122122	1685,78	1693,53	6663,55	<u>1685,52</u>	1696,06	518,02	-0,02	0,15	12,86	<u>1685,52</u>	1696,32	510,90	-0,02	0,16	13,04
122212	1449,62	1452,69	2628,61	<u>1449,03</u>	1452,55	232,05	-0,04	-0,01	11,33	<u>1449,46</u>	<u>1451,84</u>	234,19	-0,01	-0,06	11,22
122222	1082,46	1083,23	4365,28	1082,46	1084,65	452,85	0,00	0,13	9,64	1082,46	1084,68	445,40	0,00	0,13	9,80
123112	1953,12	1959,63	2617,94	<u>1946,82</u>	1959,68	240,63	-0,32	0,00	10,88	<u>1947,34</u>	<u>1956,49</u>	250,89	-0,30	-0,16	10,43
123122	1910,08	1918,55	5549,89	1910,08	1922,58	466,44	0,00	0,21	11,90	1915,80	1923,14	445,58	0,30	0,24	12,46
123212	1760,45	1762,06	2377,76	<u>1760,22</u>	<u>1761,69</u>	205,05	-0,01	-0,02	11,60	<u>1760,13</u>	<u>1761,61</u>	213,03	-0,02	-0,03	11,16
123222	1391,17	1403,60	4147,98	<u>1391,04</u>	<u>1391,74</u>	481,75	-0,01	-0,85	8,61	1391,40	1396,19	453,07	0,02	-0,53	9,16

Na Tabela 24, utilizando o “gap/MSD” como parâmetro, a versão *vRefina1_C* obteve, de 19 instâncias, três melhoras e empatou em 14 resultados. O *speedup* obteve na média 8,96, oscilando entre os valores de 0,84 à 22,18. Na versão *vRefina2_C* e utilizando também o “gap/MSD” como parâmetro, obteve-se, de 19 instâncias, duas melhoras e empatou em 16 resultados. O *speedup* obteve na média 8,82, oscilando entre os valores de 0,84 à 21,70.

Na Tabela 25, pode-se observar a comparação entre a versão sequencial com as versões paralelas *vRefina1_C* e *vRefina2_C* para o Banco de Teste 2. Para a versão *vRefina1_C*, utilizando o “gap/MSD” como parâmetro, de 30 instâncias, este último apresentou nove melhoras e empatou em 18 resultados. O *speedup* obteve na média 6,45, oscilando entre os valores de 2,02 à 10,16. A versão *vRefina2_C* obteve no parâmetro “gap/MSD” sete melhoras e empatou em 19 resultados. O *speedup* obteve na média 6,39 oscilando entre os valores de 1,89 à 10,39.

Nos testes realizados para comparação da versão sequencial com as versões *vRefina1_C* e *vRefina2_C* para o Banco de Teste 3 (Tabela 26), a versão *vRefina1_C*, utilizando o “gap/MSD” como parâmetro, de 36 instâncias, apresentou 13 melhoras e empatou em 21 resultados. O *speedup* obteve na média 11,56, oscilando entre os valores de 8,61 à 14,76. A versão *vRefina2_C* obteve no parâmetro “gap/MSD” nove melhoras e empatou em 19 resultados. O *speedup* obteve na média 11,55 oscilando entre os valores de 9,16 à 14,74.

Um resumo destes resultados, considerando todas as versões paralelas na execução com 32 *threads*, pode ser visualizado nas Figuras 28 e 29. A versão *vRefina2_C* obteve os melhores *gaps* de todas as versões para Banco de Teste 1 (Figura 28). O *speedup* apresentado pelas versões *vRefina1_C* e *vRefina2_C* foram parecidos com as versões *vRefina1_CD* e *vRefina2_CD* (Figura 29).

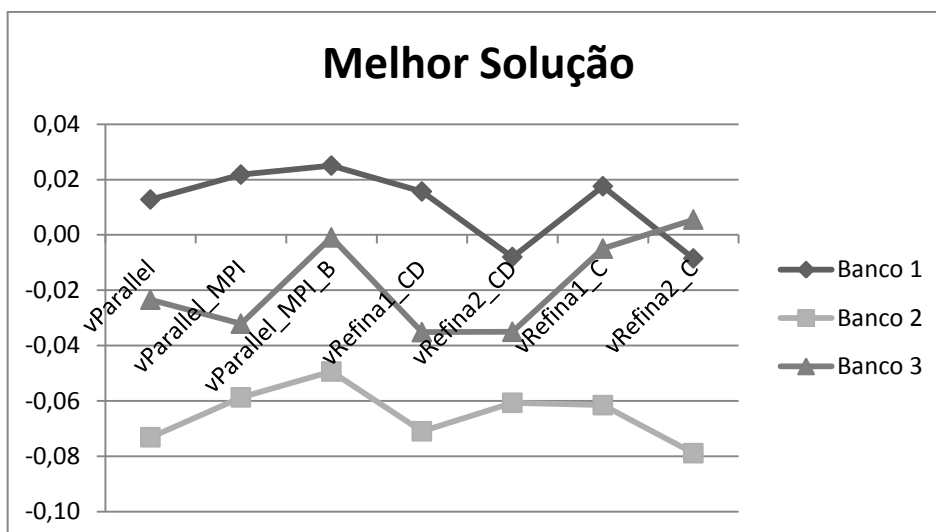


Figura 28: Gráfico Melhor Solução – Versões *vParallel*, *vParallel_MPI*, *vParallel_MPI_B*, *vRefina1_CD*, *vRefina2_CD*, *vRefina1_C* e *vRefina2_C* – 32 threads

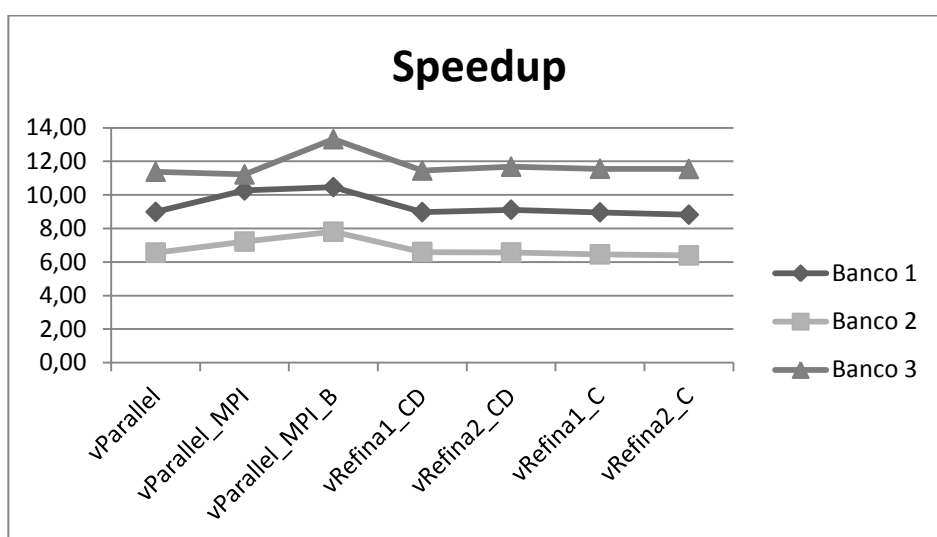


Figura 29: Gráfico Melhor Solução – Versões *vParallel*, *vParallel_MPI*, *vParallel_MPI_B*, *vRefina1_CD*, *vRefina2_CD*, *vRefina1_C* e *vRefina2_C* – 32 threads

Na Figura 28, observa-se que apenas as versões *vRefina2_CD* e *vRefina2_C* atingem a média de *gaps* negativos, para o Banco de Teste 1. Considerando os Bancos de Testes 2 e 3, as versões paralelas conseguem para quase todas as versões, médias de *gaps* negativos, com exceção da versão *vRefina2_C* que ficou ligeiramente acima para o Banco de Teste 3. Com relação ao tempo de execução, considerando os três bancos de testes, verifica-se na Figura 29,

que as versões *vRefina** não superam os *speedups* das versões que não possuem núcleos especializados, devido as *threads* que continuam refinando durante algumas iterações após o término das *threads* ILS-RVND. Uma análise da comparação destes *gaps* pode ser vista nas conclusões deste capítulo, nas Tabelas 27, 28 e 29.

Ao realizar os testes paralelos, observa-se, que várias instâncias do Bancos de Testes 1 e 2 apresentaram um *speedup* ruim comparado a outras. Um ponto comum nessas instâncias é o seu tamanho e a facilidade com que o algoritmo encontra sua solução ótima mesmo na versão sequencial. Como o objetivo de paralelizar é tirar vantagem do processamento disponível para resolver instâncias de maneira mais rápida que a versão sequencial, as instâncias foram excluídas e os dados foram novamente analisadas.

Considerando a execução com 32 *threads*, duas análises foram feitas:

- Comparam-se as melhores soluções encontradas pelas versões paralelas e a média dos resultados em 10 execuções com as melhores soluções encontradas pela versão sequencial, removendo as instâncias com um número de clientes menor que 50, com a finalidade de obter uma melhor análise do *speedup*, já que as soluções com menos de 50 clientes são de fácil resolução (tempo de execução muito rápido como pode ser visualizado nas tabelas anteriores), não havendo necessidade de paralelização.
- Comparam-se as melhores soluções encontradas pelas versões paralelas e a média dos resultados em 10 execuções com as melhores soluções encontradas pela versão sequencial, apenas para o segundo banco de teste e removendo as instâncias 100-10-1a e 100-10-1b. O objetivo deste teste é mostrar que os resultados das várias versões paralelas foram parecidos nos quesitos melhor solução e média, devido ao peso maior que estas instâncias possuem em relação às outras.

Observa-se que as instâncias com menos de 50 clientes são facilmente resolvidos pela versão sequencial. Para analisar melhor a paralelização em instâncias que realmente justificam o método, remove-se aquelas com menos de 50 clientes e contabiliza-se o *speedup* médio para cada versão. Na Figura 30, pode-se observar a diferença no *speedup* para a série “Todas” (todas as instâncias) e a série “50+” (instâncias com mais de 50 clientes). É possível verificar que houve um ganho de aproximadamente três pontos em todas as versões.

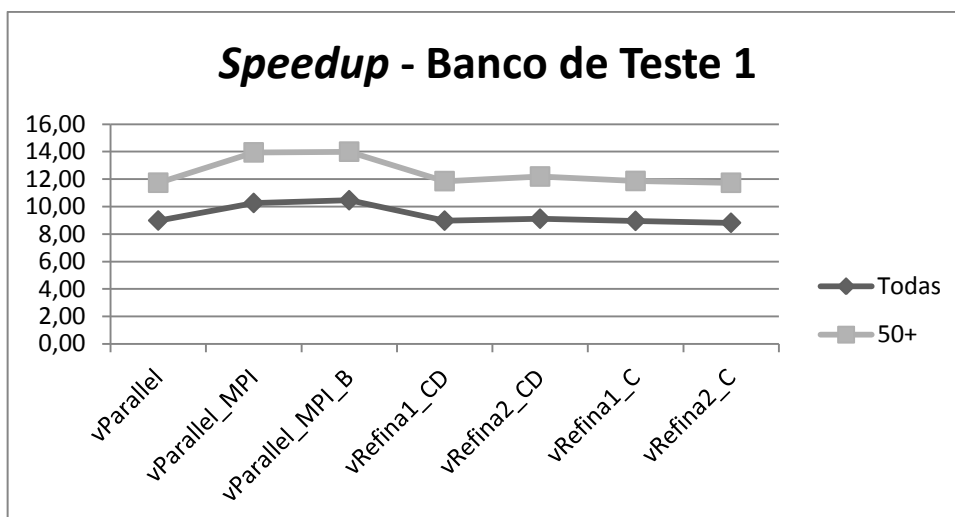


Figura 30: Gráfico *Speedup* - Instâncias com mais de 50 clientes – 32 threads – Banco de Teste 1

Para o segundo banco de teste, é possível verificar na Figura 31 que, apenas seis versões obtiveram alguma melhora que aproximadamente não passa de 0,6. Nas versões *vParallel_MPI* e *vRefina2_CD* praticamente não houve mudança. Isso ocorre pelo fato do segundo banco de testes possuir apenas quatro instâncias com menos de 50 clientes. Como o segundo banco de testes possui 30 instâncias, o fator de impacto da remoção destas quatro instâncias é bem pequeno.

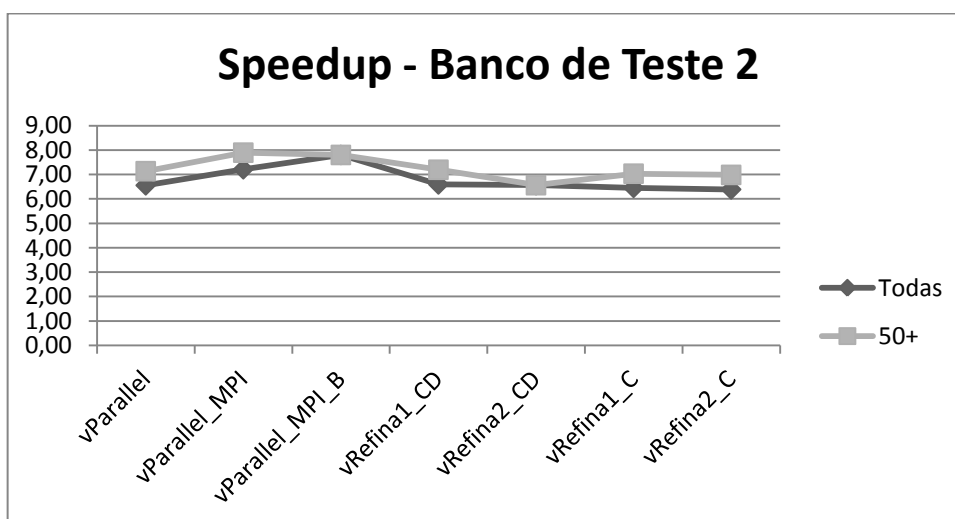


Figura 31: Gráfico *Speedup* - Instâncias com mais de 50 clientes – 32 threads - Banco de Teste 2

No segundo banco de teste foi identificado *gaps* maiores para as instâncias 100-10-1a e 100-10-1b, como por exemplo no teste da versão *vRefina1_CD*,

chegando a atingir os valores de -0,62 e -1,94 respectivamente contra o terceiro maior *gap* com -0,11 na instância 200-10-1b. Este comportamento também pode ser observado nos testes das outras versões. Como consequência, estas duas instâncias possuem um grande peso sobre o resultado quando avaliação é feita através da media do *gap* da melhor solução. A nova análise consiste em calcular novamente a média do *gap* da melhor solução sem as instâncias 100-10-1a e 100-10-1b. O resultado pode ser observado na Figura 32, onde as versões *vParallel_MPI*, *vRefina2_CD* e *vRefina1_C* passam a ter resultados competitivos com o sequencial. As versões *vParallel_MPI_B* e *vRefina1_CD*, passam a ter os piores resultados. A versão *vParallel* apresenta um resultado inferior em relação ao sequencial e a versão *vRefina2_C* continua sendo a que obteve os melhores resultados, ainda que com uma diferença ínfima do sequencial.

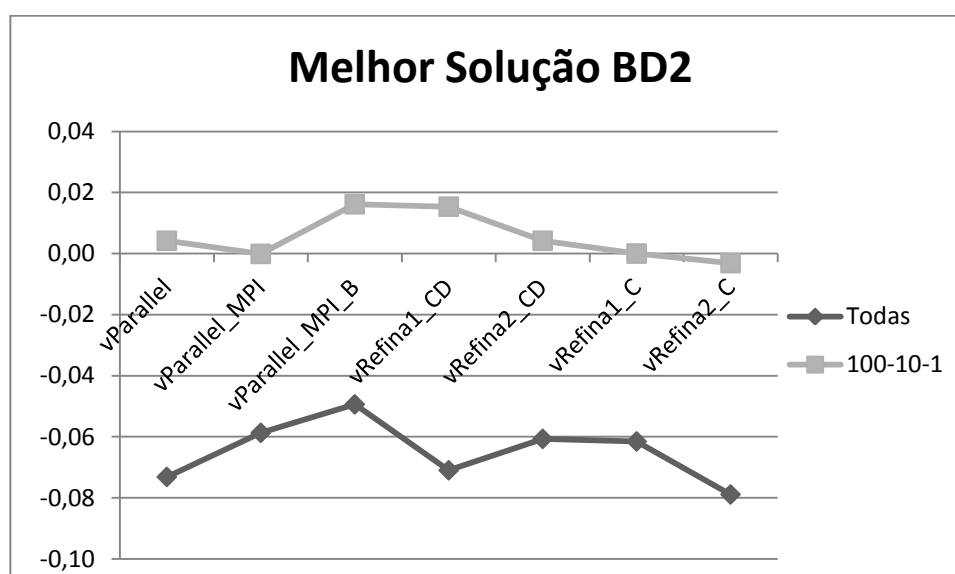


Figura 32: Gráfico Melhor Solução – Sem Instâncias 100-10-1a e 100-10-1b – 32 threads - Banco de Teste 2

6.3 Análise Experimental - Conclusões

Inicialmente, o objetivo da paralelização do algoritmo ILS-RVND era diminuir o tempo de execução e manter a qualidade das soluções em relação à versão sequencial. A primeira versão paralela, denotada *vDistrib*, consistiu em apenas dividir o trabalho do algoritmo sequencial. Os testes iniciais constataram que foi possível chegar a resultados tão bons em um tempo computacional bem menor. A partir deste ponto, foram implementadas versões paralelas com comunicação, na qual a versão mais relevante para o trabalho é a que permite comunicação entre máquinas através da biblioteca MPI. Neste caso, a comunicação é importante por permitir criar variações da ideia do algoritmo sequencial com troca de soluções entre diferentes núcleos de processamento. Além disso, o paralelismo possibilita destinar alguns núcleos para um processamento especializado, como no caso das etapas de refinamento do ILS-RVND.

Na análise inicial, realizada no *cluster* LAPORTE, foi possível observar que todas as versões paralelas propostas tiveram um bom desempenho em termos de soluções obtidas e também no critério *speedup*. Com estes resultados, a versão escolhida para testes mais detalhados foi a versão *vParallel_MPI* (com troca de mensagens entre máquinas) que foi executada no cluster OSCAR com um maior número de nós. Os resultados do cluster OSCAR e LAPORTE foram compatíveis e o objetivo passou a ser criar novas versões que pudessem não somente melhorar o *speedup* mas também melhorar a qualidade das soluções encontradas. Estas novas versões não seguem totalmente a hierarquia do algoritmo sequencial, na qual algumas *threads* passam a apenas refinar. Assim, foram configuradas as versões *vRefina1_CD*, *vRefina2_CD*, *vRefina1_C* e *vRefina2_C*, como explicado anteriormente neste capítulo.

A seguir, pode ser visualizada uma compilação das melhores soluções obtidas para os três bancos de testes usados, na execução com 32 *threads* no *cluster* OSCAR. As Tabelas 27, 28 e 29 apresentam estes resultados, onde é indicado se a versão atingiu o melhor resultado encontrado e caso contrário o quão distante ela ficou deste valor (*gap*).

Tabela 27: Melhor Solução – Banco de Teste 1

Instâncias	Melhor Solucao	Sequencial	vParallel	vParallel_MPI	vParallel_MPI_B	vRefina1_CD	vRefina2_CD	vRefina1_C	vRefina2_C
Christofides69-50x5	565,62	x	x	x	x	x	x	x	x
Christofides69-75x10	844,40	0,12	0,02	x	0,02	x	x	x	x
Christofides69-100x10	833,43	x	x	x	x	x	x	x	x
Daskin95-88x8	355,78	x	x	x	x	x	x	x	x
Daskin95-150x10	43919,90	0,01	x	x	0,01	x	x	x	0,01
Gaskell67-21x5	424,90	x	x	x	x	x	x	x	x
Gaskell67-22x5	585,11	x	x	x	x	x	x	x	x
Gaskell67-29x5	512,10	x	x	x	x	x	x	x	x
Gaskell67-32x5	562,22	x	x	x	x	x	x	x	x
Gaskell67-32x5	504,33	x	x	x	x	x	x	x	x
Gaskell67-36x5	460,37	x	x	x	x	x	x	x	x
Min92-27x5	3062,02	x	x	x	x	x	x	x	x
Min92-134x8	5709,00	x	x	0,18	0,18	0,18	0,18	0,18	0,26
Perl83-12x2	203,98	x	x	x	x	x	x	x	x
Perl83-55x15	1112,06	x	x	x	x	x	x	x	x
Perl83-85x7	1622,50	x	x	x	x	x	x	x	x
Perl83-318x4A	558415,29	0,47	0,30	0,42	x	0,10	0,44	0,85	0,09
Perl83-318x4B	669281,71	0,18	0,69	0,59	1,04	0,78	x	0,07	0,26
Or76-117x14	12296,93	x	x	x	x	x	x	x	x

Tabela 28: Melhor Solução – Banco de Teste 2

Instancia	Melhor Solução	Sequencial	vParallel	vParallel_MPI	vParallel_MPI_B	vRefina1_CD	vRefina2_CD	vRefina1_C	vRefina2_C
20-5-1a	54793	X	x	x	x	x	x	x	x
20-5-1b	39104	X	x	x	x	x	x	x	x
20-5-2a	48908	X	x	x	x	x	x	x	x
20-5-2b	37542	X	x	x	x	x	x	x	x
50-5-1a	90111	X	x	x	x	x	x	x	x
50-5-1b	63242	X	x	x	x	x	x	x	x
50-5-2a	88298	X	x	x	x	x	x	x	x
50-5-2b	67308	X	x	x	x	x	x	x	x
50-5-2BIS	84055	X	x	x	x	x	x	x	x
50-5-2bBIS	51822	X	x	x	x	x	x	x	x
50-5-3a	86203	X	x	x	x	x	x	x	x
50-5-3b	61830	X	x	x	x	x	x	x	x
100-5-1a	275079	0,14	x	x	0,07	0,14	x	x	x
100-5-1b	213568	X	0,15	0,15	0,19	0,15	0,15	x	0,10
100-5-2	193671	x	x	x	x	x	x	x	x
100-5-2b	157095	0,03	0,03	x	0,01	0,03	0,02	0,02	0,03
100-5-3a	200079	x	0,06	x	x	0,06	0,05	x	x
100-5-3b	152441	x	x	x	x	x	x	x	x
100-10-1a	294008	1,90	0,63	0,86	0,68	x	1,07	1,20	1,26
100-10-1b	234531	1,60	0,60	0,91	0,92	0,99	0,53	0,49	x
100-10-2a	243590	x	x	x	x	x	x	x	x
100-10-2b	203988	x	x	x	x	x	x	x	x
100-10-3a	252590	0,12	0,12	0,01	0,14	0,12	0,12	x	0,12
100-10-3b	204567	0,05	0,03	x	0,03	x	x	x	x
200-10-1a	475472	0,11	x	0,08	0,14	0,34	0,03	0,28	0,12
200-10-1b	375700	0,17	0,22	x	0,25	0,06	0,10	0,10	0,04
200-10-2a	448540	0,08	0,04	0,03	0,07	0,05	0,09	x	x
200-10-2b	373696	0,10	0,06	0,07	0,08	0,12	x	0,14	0,02
200-10-3a	470540	x	0,23	0,48	0,22	0,15	0,40	0,28	0,24
200-10-3b	362358	0,05	0,02	0,02	0,08	0,04	x	0,03	0,09

Tabela 29: Melhor Solução – Banco de Teste 3

Instancia	Melhor Solução	Sequencial	vParalel	vParalel_MPI	vParalel_MPI_B	vRefina1_CD	vRefina2_CD	vRefina1_C	vRefina2_C
111112	1467,68	x	x	x	x	x	x	x	x
111122	1448,37	x	x	x	x	x	x	x	x
111212	1394,80	x	x	x	x	x	x	x	x
111222	1432,29	x	x	x	x	x	x	x	x
112112	1167,16	x	x	x	x	x	x	x	x
112122	1102,24	x	x	x	x	x	x	x	x
112212	791,66	x	x	x	x	x	x	x	x
112222	728,30	x	x	x	x	x	x	x	x
113112	1238,49	x	x	x	x	x	x	x	x
113122	1245,31	x	x	x	x	x	x	x	x
113212	916,05	x	x	x	x	x	x	x	x
113222	1018,29	x	x	x	x	x	x	x	x
131112	1895,83	0,26	0,13	0,24	0,26	0,26	x	x	0,57
131122	1820,12	x	x	0,14	0,01	0,01	0,14	0,20	0,22
131212	1960,02	0,41	0,26	0,26	0,25	0,26	0,26	0,26	x
131222	1792,77	0,17	x	x	x	x	x	x	0,48
132112	1443,32	x	x	x	x	x	x	x	x
132122	1429,30	x	x	x	0,01	x	x	0,01	0,13
132212	1208,54	x	x	x	x	x	x	x	x
132222	924,68	0,18	0,03	x	0,04	x	x	0,23	x
133112	1694,18	x	x	x	x	x	x	x	x
133122	1392,01	x	x	x	x	x	x	x	x
133212	1198,08	0,67	0,67	0,67	x	0,67	0,67	0,67	0,67
133222	1151,80	x	x	x	x	x	x	x	x
121112	2237,71	0,09	x	x	x	x	x	0,07	x
121122	2137,45	0,01	0,10	0,11	0,10	x	0,10	0,01	0,02
121212	2203,53	0,02	0,01	x	0,25	x	x	0,30	0,01
121222	2214,86	0,16	0,16	x	0,18	x	x	0,09	0,27
122112	2070,43	0,18	0,09	0,09	0,18	x	0,12	0,12	x
122122	1685,52	0,02	x	0,01	x	x	x	x	x
122212	1449,03	0,04	0,04	0,09	0,03	0,04	x	0,01	0,03
122222	1082,46	x	x	x	x	x	x	x	x
123112	1946,53	0,34	0,15	x	0,09	0,03	0,02	0,12	0,04
123122	1910,08	x	0,03	0,03	x	x	x	0,30	0,30
123212	1760,13	0,02	0,02	0,02	0,02	0,03	0,01	x	x
123222	1390,87	0,02	0,04	0,02	x	0,02	0,01	0,01	0,04

Pode-se confirmar, através da Tabela 27, a facilidade que os algoritmos sequenciais e os paralelos possuem para resolver as instâncias do Banco de Teste 1. Com exceção das instâncias Perl318x4A e Perl318x4B, a diferença entre uma versão que não atingiu a melhor solução encontrada não passa de 0,18. Devido à natureza combinatória do problema e considerando a capacidade computacional atual, as instâncias Perl318x4A e Perl318x4B possuem 318 clientes e 4 depósitos, o que, conseqüentemente, torna essas instâncias de difícil resolução. Usando essas duas instâncias como parâmetro, pode-se observar melhor as diferenças entre as versões paralelas propostas. Destaca-se a versão *vRefina2_CD* que além de obter um melhor resultado para a instância Perl318x4B, alcança a melhor média dos *gaps*. Esse resultado era esperado, já que esta versão tem dois nós por máquina dedicados a fazer busca locais apenas entre clientes e depósitos e todos os depósitos dessas instâncias são utilizados em sua resolução. Analisando a Figura 28, verifica-se que as versões *vRefina2_CD* e *vRefina2_C*, são as melhores para o Banco de Teste 1, apresentando de *gaps* negativos.

Na Tabela 28 é possível visualizar que os algoritmos propostos têm facilidade em resolver instâncias com menos de 50 clientes. Observa-se que os algoritmos passam a encontrar diferentes soluções a partir da instância 100-5-1a. Outro ponto anteriormente citado são as instâncias 100-10-1a e 100-10-1b, que são de difícil resolução. Chega-se a obter *gaps* acima de 1,00 para essas duas instâncias, com destaque à versão sequencial que obteve *gaps* ruins de 1,90 e 1,60 respectivamente. A versão *vRefina1_C* se destaca conseguindo, de 30 instâncias, 21 dos melhores resultados de todas as versões do Banco de Teste 2.

Ainda considerando o Banco de Teste 2, todas as versões paralelas superam o algoritmo sequencial (Figura 28) apresentando média de *gaps* negativa. A versão *vRefina2_C* obtém os melhores resultados no critério melhor solução.

Analisando os resultados apresentadas para o Banco de Teste 3, na Tabela 29, apesar de todas as versões encontrarem a mesma melhor solução das instâncias 111112 à 113222, essas instâncias possuem no mínimo 100 clientes e 10 depósitos e não são de tão fácil resolução quanto as instâncias encontradas nos outros dois bancos de testes. Assim, se diminuirmos drasticamente o número de iterações, corre-se o risco dos algoritmos não mais atingirem essas soluções. Devido às características do Banco de Teste 3 de possuir 100, 150 ou 200 clientes, 10 ou 20 depósitos e a capacidade dos depósitos serem ilimitadas, este banco tem a

característica de possuir uma gama maior de combinações de rotas, consequentemente privilegiando as versões paralelas que possuíam nós especializados em refinar clientes e depósitos, com destaque para as versões *vRefina1_CD* e *vRefina2_CD*, que de 36 instâncias obtiveram 28 das melhores soluções obtidos por todas as versões. As versões com núcleos especializados em fazer apenas busca locais foram incentivadas principalmente pelas características do banco de teste 3. As versões *vRefina1_CD* e *vRefina2_CD* são as que alcançam os melhores resultados na média dos *gaps* (Figura 28).

Considerando os resultados obtidos pelas versões paralelas é possível observar que dependendo da característica da instância uma determinada versão pode ser mais vantajosa ou não, como por exemplo as versões com threads especializadas em busca locais em clientes que são melhores para instâncias com um pequeno número de depósitos e um grande número de clientes. Assim, uma ideia interessante seria analisar previamente as características de uma instância e disparar a versão que melhor se adapta para resolver o problema.

No próximo capítulo, apresentam-se as conclusões finais e sugestão das etapas futuras desta tese.

Conclusões e Etapas Futuras

O PLR é um problema estudado há 49 anos com aplicações práticas nas mais diversas áreas. Considerado a combinação dos problemas de Localização de Facilidades com o Problema de Roteamento. O estudo do PLR de dois níveis é o mais estudado por ser o mais comum quando considerado um problema real. Ao longo dos anos, vários trabalhos foram publicados com propostas de soluções para os mais diversos problemas teóricos e estudos de casos, análise e artigos de revisão.

Podemos dividir os métodos de resolução em dois grupos: exatos e heurísticos. Os métodos exatos bem sucedidos só resolvem problemas com até 100 clientes deixando uma gama de instâncias para serem resolvidos pelos métodos heurísticos. Apesar de vários métodos tentarem particionar o problema em dois subproblemas, devido à sua natureza, os melhores métodos foram aqueles que consideram a informação do roteamento para realizar a localização iterativamente, realizando correções nas duas tomadas de decisão

O método ILS-RVND é conhecido na literatura por resolver problemas de roteamento com grande eficiência. Este método foi adaptado para o PLR demonstrando algumas deficiências que foram sendo corrigidas ao longo de estudos. A primeira deficiência detectada, como é de conhecimento dos autores, é quando o somatório da demanda tem valores próximos à soma da capacidade dos depósitos ótimos da solução. Este problema foi corrigido através de um cálculo prévio que decide quando utilizar uma construção baseado em *First Fit Decreasing*. Outro problema era o tempo de execução em relação a outros trabalhos publicados ser muito superior.

Apesar de melhores soluções, o tempo chegava a ser 10 vezes maior que os resultados publicados. Isso é consequência dos autores terem conhecimento sobre suas instâncias propostas por ele mesmos, obtendo resultados satisfatórios através de uma busca direcionada as características da instância. Pode-se observar que, para outros bancos de testes os resultados caíam de rendimento drasticamente. Uma comparação mais “justa” entre algoritmos pode ser realizada com a publicação do relatório técnico de Hemmelmayr (2011). Os resultados são robustos para o três

bancos de testes (incompleto para o banco de teste de “Barreto”) e os tempos de execução aumentaram drasticamente, como já se era esperado. Mesmo com a publicação de Hemmelmayr, duas otimizações foram realizadas na proposta inicial do ILS-RVND. Foi detectado que as perturbações entre depósitos são as mais pesadas por mexerem com várias rotas e clientes. Outra conclusão é que não é necessário serem realizadas tantas vezes. Após a perturbação de um depósito, é desejado que várias buscas e perturbações entre clientes sejam realizadas. Com esses dados em mãos, a probabilidade de ocorrer essas perturbações diminuíram para 50% em relação as perturbações entre clientes gerando economia de tempo de 25% e melhorando a qualidade das soluções. A consequência desse estudo é que o algoritmo já encontrava o número de veículos ótimos com suas rotas relacionadas otimamente ao depósitos ótimos. O que faltava era os clientes estarem nas rotas e posições ótimas. Para corrigir as rotas, foi inserida uma segunda fase de refinamento.

Quanto aos três bancos de testes escolhidos para estudo (“BARRETO”, “PRINS” e “TUZUN”), eles apresentam algumas diferenças. Diferentemente dos bancos de testes de “PRINS” e “TUZUN”, as instâncias de “BARRETO” não possuem custo por veículo utilizado. O banco de teste de “BARRETO” possui uma instância retirada de um estudo de caso de distribuição de sangue na cidade de Chicago (OR e PIERSKAL, 1979). No banco de “TUZUN”, os custos dos depósitos são muito menores que o custo do roteamento variando entre $1/5$ a $1/3$ do custo global. Além disso, no banco de teste de “TUZUN” todos os depósitos são capazes de suprir todas as demandas, ou seja, existem soluções com apenas um depósito localizado para todas as instâncias.

A heurística proposta ILS-RVND obteve resultados satisfatórios. Usando a média dos *gaps* das melhores soluções comparadas com as melhores soluções da literatura como critério, o algoritmo obteve -0,56 para o banco de teste “BARRETO”, 0,28 para “PRINS” e -0,30 para “TUZUN”. Novas soluções foram encontradas para os três bancos de testes e quando comparado com algoritmo ALNS, proposto por Hemmelmayr (2011), a média do “gap/MS” é -0,11 no banco de teste “PRINS” e -0,40 para “TUZUN”. Durante o estudo de parametrização do algoritmo e estudo das probabilidades das perturbações, concluiu-se que ainda pode-se obter melhores resultados configurando o algoritmo para cada banco de teste. Uma maneira de

obter estes melhores resultados sem utilizar este artifício é a implementação de versões paralelas.

Inicialmente três versões paralelas foram implementadas: *vDistrib* – sem comunicação; *vParallel* – com comunicação através da memória compartilhada; e *vParallel_MPI* – com comunicação entre máquinas através da biblioteca MPI. Como apresentado no Capítulo 6, os experimentos comprovaram que estas versões foram tão boas quanto o algoritmo sequencial, porém encontrando suas soluções em tempos computacionais bem menores.

A média do *speedup* para as versões *vDistrib*, *vParallel* e *vParallel_MPI*, considerando 16 *threads*, nos testes iniciais realizados no *cluster* LAPORTE foi, respectivamente: banco de teste 1 – 8,27, 6,36, 4,96; banco de teste 2 – 5,62, 4,39, 4,17; e banco de teste 3 – 5,42, 5,35, 6,03.

Dados os resultados promissores dos testes iniciais, uma segunda etapa de testes consistiu em criar versões paralelas que possuíssem configurações diferentes do algoritmo sequencial, com a especialização de algumas *threads* em etapas de refinamento diferenciadas. Nestes casos, o objetivo passou a ser não somente melhorar o *speedup*, mas também a qualidade das soluções. Estas novas versões se tornam possíveis por serem executadas em uma plataforma que fornece vários núcleos de processamento (*cores*) e assim, alguns núcleos poderem executar *threads* especializadas em diferentes etapas do algoritmo ILS-RVND e trocarem informações entre si através da biblioteca MPI. São estas versões: *vRefina1_CD*, *vRefina2_CD*, *vRefina1_C* e *vRefina2_C*.

No critério *speedup* estas novas versões obtiveram desempenho competitivos com a versão *vParallel_MPI*. Considerando a execução no *cluster* OSCAR com 32 *threads*, a média do *speedup* para as versões *vParallel_MPI*, *vRefina1_CD*, *vRefina2_CD*, *vRefina1_C* e *vRefina2_C* foi, respectivamente: banco de teste 1 – 10,26, 8,97, 9,11, 8,95, 8,81; banco de teste 2 – 7,21, 6,59, 6,56, 6,45, 6,39; e banco de teste 3 – 11,21, 11,45, 11,68, 11,55, 11,54.

De uma maneira geral, os testes paralelos tornaram possível execuções com um grande número de iterações conseguindo melhorar os resultados já obtidos pela versão sequencial em um tempo hábil. Um bom exemplo é a instância Perl318x4A, que ao dobrar o número de iterações, demorava 110 minutos para executar na versão sequencial, enquanto que na versão *vRefina2_C* com 32 *threads* foi gasto um tempo médio de 8,1 minutos. Outro ponto positivo do paralelismo é

poder possuir diferentes nós realizando diferentes combinações de busca locais ou diferentes etapas do algoritmo proposto. Isso possibilitou encontrar melhores soluções que a versão sequencial.

Uma abordagem interessante para combinar essas versões seria criar uma plataforma que tentaria identificar as características da instância a ser resolvida e a execução seria direcionada para a versão que melhor resolvesse esta instância. Com as versões disponíveis neste trabalho, poderíamos montar um programa com a seguinte configuração: uma combinação das versões *vRefina1_CD* com a *vRefina2_C* ou *vRefina2_CD* com *vRefina2_C*. As versões *vRefina1_CD* e *vRefina2_CD* são as melhores versões com o terceiro banco de teste e a *vRefina2_C* é a melhor tanto para o primeiro quanto para o segundo banco de testes.

Um dos testes que não foi possível realizar neste trabalho, devido as limitações de disponibilidade do *cluster* OSCAR e tempo, foi verificar até que ponto é vantajoso aumentar o número de máquinas executando os algoritmos paralelos. Além disso, realizar testes com soluções alvo e testes com tempo limite de execução.

Outras versões paralelas a serem futuramente testadas são aquelas com núcleos de refinamentos diferentes: um refinando apenas clientes e outro refinando clientes e depósitos.

os.

Como estudo ainda a ser realizado, deseja-se analisar as modelagens matemáticas para o problema a fim de encontrar novas soluções ótimas, se possível, e propor um método híbrido combinando o ILS-RVND proposto exato de resolução. Prevê-se que as soluções para as instâncias que não foram encontradas com a construção *First Fit Decreasing* sejam encontradas pelo método híbrido.

Referências

- Alba, E., Luque, G., Nesmachnow, Sergio., “Parallel metaheuristics: recent advances and new trends”. *Intl. Trans. in Op. Res.*, páginas 1–48, 2012.
- Averback, I., Berman, O. “Routing and Location Routing p-Delivery Men Problems on a Path”. *Transportation Science* 28, páginas 162-166, 1994.
- Averback, I., Berman, O. “Probabilistic Sales-Delivery Facility Location Problems”. *Transportation Science* 29, páginas 184-197, 1995.
- Balakrishnan, A., Ward, J. E., Wong, R. T. “ Integrated Facility Location and Vehicle Routing Models: Recent Work and Future Prospects”, *American Journal of Mathematics and Management Sciences* 7, páginas 35-61, 1987.
- Baldacci, R., Mingozzi, A., Wolfler Calvo, R. “An Exact Method for the Capacitated Location-Routing Problem”. *Operations Research* 59, 2011.
- Barreto, S. S., “Análise e Modelização de Problema de Localização-Distribuição” *Tese de Doutorado, Universidade de Aveiro, Portugal*, 2004.
- Barreto, S. S. Ferreira C., Paixão, J., Santos, B. S., “Using Clustering Analysis in a Capacitated Location-Routing Problem”. – *European Journal of Operational Research* 179, páginas 968-977, 2007.
- Baumol, W. J. e Wolfe, P. “A Warehouse-Location Problem” *Operations Research* 6, páginas 252-263, 1958.
- Belenguer, J. M., Benavent, E., Prins, C., Prodhon, C., Wolfler-Calvo, R., “A Branch-and-Cut Algorithm for the Location-Routing Problem with Depot and Vehicle Capacities”. *Computers & Operations Research*, 2010.
- Belenguer, J. M., Benavent, E., Prins, C., Prodhon, C., Wolfler-Calvo, R., “A Branch-and-Cut Method for the Capacited Location-Routing Problem”. *Computers and Operations Research* 38, páginas 931-941, 2011.
- Bellman, R. “On a Routing Problem”. *Quarterly of Applied Mathematics* 16, páginas 87-90, 1958.
- Berger, R. T. “Location-Routing Models for Distribution System Design”. *Tese de Doutorado, Northwestern University, Evanston, Illinois*, Dezembro, 1997.
- Berman, O., Jaillet, P., Simchi-Levi, D. “Location-Routing Problems with Uncertainty”. *Facility Location: A Survey of Applications and Methods*, páginas 427-453, 1995.

- Berman, O., Simchi-Levi, D. "Finding the Optimal a Priori Tour and Location of a Traveling Salesman with Nonhomogeneous Customers". *Transportation Science* 22, páginas 148-154, 1988.
- Berman, O., Simchi-Levi, D. "Minisum Location of Travelling Salesman". *Networks* 16, páginas 239-254, 1986.
- Bestimas, J., Jaillet, P., Odoni, A. R. "A Priori Optimization". *Operations Research* 38, páginas 1019-1033, 1990.
- Boffey, B., Karkazis, J. "Models and Methods for Location and Routing Decision Realating to Hazardous Materials", *Studies in Location Analysis* 5, páginas 146-166, 1993.
- Bookbinder, J., Reece, K. E. "Vehicle Routing Considerations in Distribution Systems Design". *European Journal of Operational Research* 37, páginas 204-213, 1988.
- Burness, R. C., White, J. A. "The Traveling Salesman Location Problem", *Transportation Science* 10, páginas 348-360, 1976.
- Bruns, A., Klose, A. "A "Locate First – Route Second" Heuristic for a combined Location Routing Problems". *Proceedings of the Second International Workshop on Distribution Logistics* páginas 1-6, 1997.
- Cappanera, P., Gallo, G., Maffioli, F. "Discrete Facility Location and Routing of Obnoxious Activities"; *Discrete Applied Mathematics* 133, páginas 3-28, 2004.
- Çetiner, S., "An Iterative Hub Location and Routing Problem for Postal Delivery Systems", *Tese de Doutorado, The Middle East Technical University*, Turquia, 2003.
- Chan, A. W., Francis, R. L. "A Round Trip Location Problem on a Tree Graph". *Transportation Science* 10, páginas 35-51, 1976.
- Chan, A. W., Hearn, D. W. "A Rectilinear Distance Round Trip Location Problem". *Transportation Science* 11, páginas 107-123, 1977.
- Chan, Y., Carter, W. B., Burness, M. D. "Multi-Depot, Multi-Vehicle, Location-Routing Problem with Stochastically Processed Demands". *Computers and Operations Research* 28, páginas 803-826, 2001.
- Chien, T. W. "Heuristic Procedure for Practical Sized Uncapacited Location-Capacitated Routing Problems" *Decisions Sciences* 24, páginas 995-1021, 1993.
- Christofides, N., Eilon, S. "Expected Distances in Distribution Problem". *Operational Research Quarterly* 20, páginas 437-443, 1969.
- Clarke, G. e Wright, J. "Scheduling of Vehicles from a central depot to a number of delivery points". *Operations Research* 12, páginas 586-581, 1964.

Contardo, C., Cordeau, J., Gendron, B. "A Branch-and-Cut Algorithm for the Capacitated Location-Routing Problem". *Proceedings of the VI ALIO/EURO Workshop on Applied Combinatorial Optimization*, Buenos Aires, Argentina, 2010.

Cooper, L. "Heuristic Methods for Location-Allocation Problems". *SIAM Review* 6, páginas 37-53, 1964.

Coutinho-Rodrigues, J., Clímaco, J., Current, J., Ratick, S. "An Interactive Spatial Decision System for Multiobjective HAZMAT Location-Routing Problem", *Studies in Location Analysis* 9, páginas 39-42, 1997.

Daskin, M. S. "Location, Dispatching and Routing Models for Emergency Services with Stochastic Travel Times". *Spatial Analysis and Location-Allocation Models*, páginas 224-265, 1987.

Dejax, P. J. "A Methodology for Warehouse Location and Distribution Systems Planning". *Proceedings of the 2nd International Seminar on Freight Transport Planning and Logistics*, 1988.

Derks, J. J. M., Spijksman, F. C. R. "A Decision Support System for Location Facilities and Routing Traffic on a Factory Site". <http://www.edocs.unimaas.nl/files/rep96009.pdf>, 1996.

Dongarra, J., Foster, I., Fox, G., Gropp, W., Kennedy, K., Torczon, L., White, A., "Source Book of Parallel Computing", *Morgan Kaufmann, Primeira Edição*, 2002.

Drezner, Z., Wesolowsky, G. O. "A Trajectory Approach to the Round-Trip Location Problem". *Transportation Science* 16, páginas 56-66, 1982.

Drezner, Z. "Fast Algorithms for the Rectilinear Round-Trip Location Problem". *IIE Transactions* 14, páginas 243-248, 1982.

Drezner, Z. "O(N log n) Algorithm for the Rectilinear Round Trip Location-Problem". *Transportation Science* 19, páginas 91-93, 1985.

Drezner, Z., Steiner, G., Wesolowsky, G. O. "One Facility Location with Rectilinear Tour Distance". *Naval Research Logistics Quarterly* 32, páginas 391-405, 1985.

Drezner, Z., "Facility Location: A survey of Applications and Methods". *Springer Verlag, New York*, 1995.

Duhamel C., Lacomme, P., Prins, C., Prodhon, C. "A Memetic Approach for the Capacitated Location Routing Problem". *International Workshop on Metaheuristics for Logistics and Vehicle Routing*, Troyes, France 2008

Eilon, S., Watson –Gandy, C. D. T. e Christofides, N. "Distribution Management: Mathematical Modelling and Practical Analysis". *Charles Griffin and Company Limited, London*, 1971.

- Eiselt, H. A., Laporte, G. "Integrated Planning in Distribution System" *International Journal of Physical Distribution and Materials Management* 19, páginas 14-19, 1989.
- Ferreira, C. M. S. "Problemas de Localização e Distribuição Mutlicritério – Aproximações e Estudo de Alguns Casos com Implicações Ambientais". *Tese de Doutorado, Universidade de Aveiro, Aveiro, Portugal*, 1997.
- Fisher, M. L. e Jailkumar, R. "A Generalized Assignment Heuristic for Vehicle Routing". *Networks* 11, páginas 109-124, 1981.
- Francis, R. L., McGinnis, L. F. e White, J. A. "Locational Analysis. *European Journal of Operational Research* 12, páginas 220-252, 1983.
- Geer, D. "Chip Makers turn to multicore processors", *Computer* 38, número 5, páginas 11-23, 2005.
- Giannikos, I. "A Multiobjective Programming Model for Locating Treatment Sites and Routing Hazardous Wastes". *European Journal of Operational Research* 104, páginas 333-342, 1998.
- Ghiani, G., Laporte G., "Eulerian Location Problems", *Networks* 34, páginas 291-302, 1999.
- Ghiani, G., Laporte, G., "Locatoin-Arc Routing Problems". *OPSEARCH* 38, páginas 151-159, 2001.
- Ghosh, J. K., Sinha, S. B., Acharya, D. "A Generalized Reduced Gradient Based Approach to Round-Trip Location Problem". *Scientific Management of Transport Systems*, páginas 209-213, 1981.
- Griffiths, D. O. "The use of regression analysis in a depot location exercise". *Applied Statistics* 17, páginas 57-63, 1968.
- Haley, K. B. "The Siting of Depots". *International Journal of Production Research* 2, páginas 41-45, 1963.
- Hakimi, S. L. "Optimum Locations of Switching Centers and the Absolute Centers and Medians of a Graph". *Operations Research* 12, páginas 450-459, 1964.
- Hansen, P. H., Hegedahl, B., Hjortkjaer, S., Obel, B. "A Heuristic Solution to the Warehouse Location and Routing Problem". *European Journal of Operational Research* 76, páginas 111-127, 1994.
- Harrisson, H. "A Planning System for Facilities and Resources in Distribution Networks", *Interfaces* 9, páginas 6-22, 1979.
- Hemmelmayer, V., C., Cordeau, J., Crainic, T., G., "An Adaptative Large Neighborhood Search Heuristic for Two-Echelon Vehicle Routing Problems Arising in City Logistics" *technical Report CIRRELT-2011-42*, Université de Montréal, 2011.

Jamil, M., Batta, R., Malon, D. "The Traveling Repairperson Home base Location Problem". *Transportation Science* 28, páginas 150-161, 1994.

Jacobsen, S. K., Madsen, O. B. G. "A Comparative Study of Heuristics for a Two Level Routing-Location Problem". *European Journal of Operational Research* 5, páginas 378-387, 1980.

Jaemyung, A. "The Generalized Location Routing Problem with Profits for Planetary Surface Exploration and Terrestrial Applications", *Tese de Doutorado, Massachusetts Institute of Technology, Estados Unidos*, 2008.

Johnson, M. P., Gorr, W. L., Roehrig, S. F., Heinz, J. H. "Location/Allocation/Routing for Home Delivered Meals Provision Models and Solution Approaches". *International Journal of Industry Engineering* 9, páginas 45-56, 2002.

Ichmori, T., Nishida, T. "Note on Rectilinear Distance Round-Trip Location Problem". *Transportation Science* 19, páginas 84-91, 1985.

Intel, "From a Few Cores to Many: A Tera-scale Computing Research Overview", ftp://download.intel.com/research/platform/terascale/terascale_overview_paper.pdf , 2006.

Klose, A. "Using Clustering Methods in Problems of Combined Location and Routing". *Operations Research Proceedings*, páginas 411-416, 1995.

Kolen, A. "The Round Trip p-Center and Covering Problem on a Tree". *Transportation Science* 19, páginas 222-234, 1985.

Labbé, M., Laporte, G. "Maximizing User Convenience and Postal Service Efficiency in Post Box Location". *Belgian Journal of Operations Research* 26, páginas 21-35, 1986.

Laporte, G. "A Survey of Algorithms for Locations Routing Problems". *Inverstigacion Operativa* 1, páginas 93-123, 1989.

Laporte, G. "Location-Routing Problems". *Vehicle Routing: Methods and Studies*, páginas 163-197, 1988.

Laporte, G., Dejax, P. J. "Dynamic Location-Routing Problems". *Journal of Operatinal Research Society* 40, páginas 471-482, 1989.

Laporte, G., Louveaux, F., Mercure, H. "Models and Exact Solutions for a Class of Stochastic Location Routing Problems". *European Journal of Operational Research* 39, páginas 71-78, 1989.

Laporte, G., Nobert, Y., Arpin, D., "An Exact Algorithm for Solving a Capacitated Location Routing Problem". *Annals of Operations Research* 6, páginas 293-310, 1986.

Laporte, G., Nobert, Y., Pelletier, P. "Hamilton Locations Problems". *European Journal of Operational Research* 12, páginas 82-89, 1983.

Laporte, G., Norbert, Y., Taillefer, S. "Solving a Family of Multi-Depot Vehicle routing and Location-Routing Problems". *Transportation Science* 22, páginas 161-172, 1988.

Lawrence, R. M., Pengilly, P. "The Number and Location of Depots Required for Handling Products for Distribution to Retail Stores in South-East England" *Operational Research Quarterly* 20, páginas 23-32, 1969.

Levy, L, Bodin, L. "The Arc Oriented Location Routing Problem". *INFOR* 27, páginas 74-94, 1989.

List, G. F., Mirchandani, P. B. "An Integrated Network/Planar Multiobjective Model for Routing and Siting for Hazardous Materials and Waste". *Transportation Science* 25, páginas 146-156, 1991.

List, G. F., Mirchandani, P. B., Turnquist, M. A., Zografos, K. G. " Modeling and Analysis for Hazardous Materials Transportation: Risk Analysis Routing/Scheduling and Facility Location". *Transportation Science* 25, páginas 100-114, 1991.

Lee, Y., Kim, S., Lee, S., Kang, K. "A Location-Routing Problem in Designing Optical Internet Access with WDM Systems". *Photonic Network Communications* 6, páginas 151-160, 2003.

Lourenço, H. R., Martin, O. C., Stutzle, T. "Iterated Local Search". *Fred Glover e Gary A. Kochenberger (eds.)*, *Handbook of Metaheuristics*, Kluwer Academic Publishers, Norwell, MA, páginas 321-353, 2002.

Madsen, O. B. J. " A Survey of Methods for Solving Combined Location-Routing Problems". *Scientific Management of Transport Systems*, páginas 194-201, 1981.

Madsen, O. B. J. "Methods for Solving Combined Two Level Location Routing Problems of Realistic Dimension". *European Journal of Operational Research* 12 páginas 295-301, 1983.

Message Passing Forum, "MPI: A Message Passing Interface. Technical Report", University of Tennessee, 1995

Mladenovic, N., Hansen, P. "Variable Neighbourhood Search". *Computers & Operations Research*, v. 24, n. 11, páginas. 1097-1100, 1997.

Maranzana, F E. "On the location of supply points to minimize transportation costs". *IBM systems Journal* 2 , páginas 194-201, 1963

Maranzana, F E. "On the location of supply points to minimize transportation costs". *Operations Research Quarterly*, páginas 261-270, 1963

Min, H., "Consolidation Terminal Location-Allocation and Consolidated Routing Problems". *Journal of Business Logistics* 17, páginas 235-263, 1996.

Min, H., Jayaraman, V., Srivastava, R. "Combined Location Routing Problems: A Synthesis and Future Research Directions". *European Journal of Operational Research* 108, páginas 1-15, 1998.

Mine, M., Silva, M. S. A., Subramanian, A., Ochi, L. S., Souza, M. J. F. "A hybrid heuristic based on iterated local search and Genius, for the vehicle routing problem with simultaneous pickup and delivery". *International Journal of Logistics Systems Management (IJLSM)* – Vol. 10(2), páginas 142–157, 2011.

Murty, K. G., Djang, P. A. "The U. S. Army National Guard's Mobile Training Simulators Locations and Routing Problems". *Operations Research* 47, páginas 175-182, 1999.

Nagy, G., Salhi, S. "The Many-to-Many Location Routing Problem". *Top-Sociedad Española de Estadística e Investigación Operativa* 6, páginas 261-275, 1998.

Nagy, G., Salhi, S. "Nested Heuristic Approach for the Combined Location Vehicle Fleet Mix Problem". *Journal of the Operational Research Society* 47, páginas 1166-1174, 1996.

Nagy G., Salhi, S. " A Nested location-routing heuristic using route length estimation. *Studies in Location Analysis* 10, páginas 109-127, 1996.

Nagy, G., Salhi, S. "Location-routing: Issues, models and methods". *European Journal of Operational Research* 177, páginas 649-672, 2007.

Nambiar, J. "Location and Vehicle Routing in the Malaysian Rubber Industry". *Tese de Doutorado Katholieke Universiteit Leuven, Belgica*, Fevereiro de 1988.

Nambiar, J. M., Gelders, L. F., Wassenhove, L. N. "Plant Location and Vehicle Routing in the Malaysian Rubber Smallholder Sector: A Case Study". *European Journal of Operational Research* 38, páginas 14-26, 1989.

Or, I. "Traveling Salesman-Type Combinatorial Problems and their Relation to the Logistics of Regional Blood Banking". *Tese de Doutorado, Northwest University, Evanston, Illinois*, 1976.

Or, I., Pierskall, W. O., "A Transportation Location Allocation Model for Regional Blood Banking". *AIIE Transactions* 11, páginas 86-95, 1979.

Penna, P. H. V., Subramanian, A., Ochi, L. S., Uchoa, E. "A Hybrid Algorithm for the Heterogenous Fleet Vehicle Routing Problem". *European Journal of Operational Research - EJOR – ELSEVIER*, v. 221, páginas 285-295, (2012a).

Perl, J. "A Unified Warehouse Location-Routing Analysis". *Tese de Doutorado, Northwestern University, Evanston*, 1983.

Perl, J., Daskin, M. S. "A Unified Warehouse Location Routing Methodology". *Transportation Research B* 19B, páginas 92-111, 1984.

Perl, J., Daskin, M. S. "A Warehouse Location-Routing Problem". *Transportation Research B* 19B, páginas 381-396, 1985.

Prins, C., Prodhon, C., Wolfler Calvo, R. . "Nouveaux algorithmes pour le problème de localisation et routage sous contraintes de capacité" *MOSIM* 4, páginas 1115-1122. 2004.

Prins, C., Prodhon, C., Wolfler Calvo, R. "Solving the Capacitated Location-Routing Problem by a Grasp Complemented by a Learning Process and a Path Relinking" *4OR – Quarterly . Journal of . Operations Research* 4(3), páginas 221–238, 2006.

Prins, C., Prodhon, C., Ruiz, A., Siriano, P., Calvo, R. W. "Solving the Capacitated Location-Routing Problem by a Cooperative Lagrangean Relaxation-Granular Tabu Search Heuristic" *Transportation Science* 41, páginas . 470-483, 2007

Revelle, V., Cohon, J., Shobrys, D. "Simultaneous Siting and Routing in the Disposal of Hazardous Wastes". *Transportation Science* 25, páginas 138-145, 1991.

Ribeiro, B., "Frota das capitais quase dobra em 10 anos; SP ganha 3,4 mi de veículos", *Estadão*, <http://www.estadao.com.br/noticias/cidades,frota-das-capitais-quase-dobra-em-10-anos-sp-ganha-34-mi-de-veiculos,938682,0.htm>, acessado em Domingo, 21 de outubro de 2012.

Salhi, S. "The Integration of Routing Into the Location-Allocation and Vehicle Fleet Composition Problems". *Tese de Doutorado, University of Lencaster*, 1987.

Salhi, S., Fraser, M. "An Integrated Heuristic Approach for the Combined Location Vehicle Fleet Mix Problem", *Studies in Locational Analysis* 8, páginas 3-21, 1996.

Salhi, S., Naggy, G. "Consistency and Robustness in Location-Routing". *Studies in Location Analysis* 13, páginas 3-19, 1999.

Salhi, S., Rand, G. G. " The Effect of Ignoring Routes when Routing Depots". *European Journal of Operational Research* 39, páginas 293-295, 1987.

Shobrys, D. "A Model for Selection of Shipping Routes and Storage Locations for Hazardous Substance". *Tese de Doutorado, Johns Hopkins University, Baltimore*, 1981.

Simchi-Levi, D. "The Capacitated Traveling Salesman Location Problem". *Transportation Science* 25, páginas 9-18, 1991.

Simchi-Levi, D., Berman, O. " A Heuristic Algorithm for the Traveling Salesman Location Problem on Networks". *Transportation Science* 25, páginas 9-18, 1991.

Srivastava, R. "Algorithm for Solving the Location-Routing Problem". *Tese de Doutorado, The Ohio State University, Columbus , Ohio*, 1986.

Srivastava R. "Alternate Solution Procedures for the Location Routing Problem", *Tese de Doutorado, The Ohio State University*, 1986.

Srivastava, R., Benton, W. C. "The Location Routing Problem Considerations in Physical Distribution System Design" *Computers and Operations Research* 17, páginas 427-435, 1990.

Sterlle, C. "Location-Routing models and methods for Freight Distribution and Infomobility in City Logistics", *Tese de Doutorado*,

Stowers, C. L., Palekar, U. S., "Location Models with Routing Considerations for a Single Obnoxious Facility". *Transportation Science* 27, páginas 350-362, 1993.

Tuzun, D., Burke, L. I. "A Two-Phase Tabu Search Approach to the Location Routing Problem". *European Journal of Operational Research* 116, páginas 87-89, 1999.

Ward, J. H. "Hierarchical Grouping to Optimize an Objective Function". *Journal of the American Statistical Association* 58, pg 236-244, 1963.

Watson-Gandy, C. D. T., Dohrn, P. J. Depot Locations with Van Salesman - A Pratical Approach". *OMEGA – The International Journal of Management Science* 1, páginas 321-329, 1973.

Weber, A., "Theory of the Location of Industries". 7ª edição *The University of Chicago Press, Chicago*, 1929

Webb, M. "Cost Functions in the Locations of Depots for Multiple-Delivery Journeys". *Operational Research Quartely* 19, páginas 311-321, 1968

Wong D. W. S., Meyer, J. W. "A Spatial Decision Support System Approach to Evaluate the Efficiency of Meals on Wheels Program". *Professional Geographer* 45, páginas 332-341, 1993.

Wu, T., Low, C., Bai, J., "Heuristic Solutions to Multi-Depot Location Routing Problems". *Computers and Operations Research* 29, páginas 1393-1415, 2002.

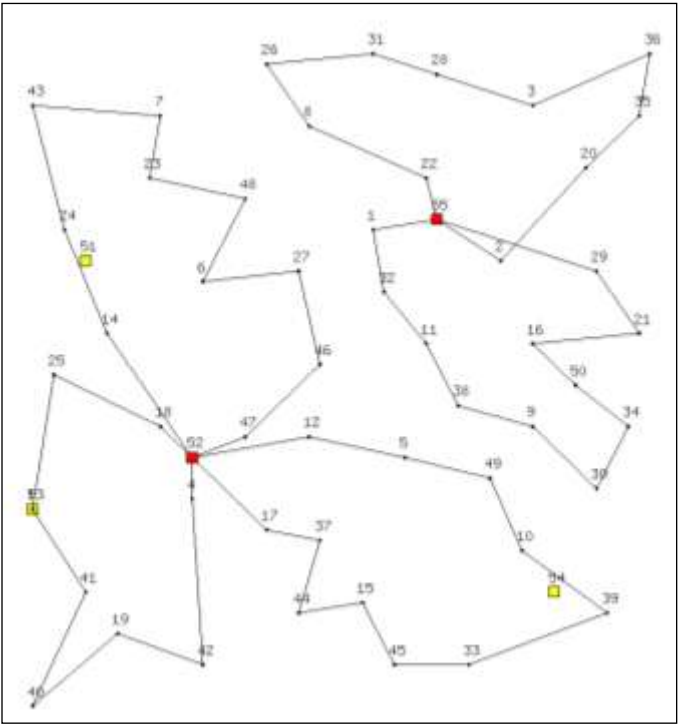
Wren, A., Holliday, A. "Computer Scheduling of Vehicles from One or More Depots to a Number of Delivery Points". *Operational Research Quarterly* 23, páginas 333-345, 1972.

Zografos, K. G., Samara, S. "Combined Location Routing Model for Hazardous Waste Transportation and Disposal". *Transportation Research Record* 1245, páginas 52-59, 1989.

Apêndice

Neste apêndice apresenta-se a melhor solução encontrada para cada instâncias pelos algoritmos presentes neste trabalho para o banco de teste “Barreto” e soluções ótima encontradas para o banco de teste “Prins”. As soluções são mostradas graficamente utilizando o modelo de desenho apresentado em Barreto (2004). Os depósitos são representados por quadrados e os clientes por pontos pretos numerados. Neste trabalho, representamos os depósitos utilizados com cinza escuro e os não utilizados com cinza claro. Os gráficos foram construídos utilizando a biblioteca gráfica OpenGL. Abaixo da representação gráfica da solução está o nome da instância. Em seguida, é encontrado o custo global da solução (a soma do custo dos depósitos utilizados mais o custo de roteamento mais o custo de utilizar cada veículo, se houver). Se a solução for comprovadamente ótima, estará sinalizada entre parênteses em seguida do valor do custo. Logo após, são apresentados as rotas em forma de vetores. Pode-se observar que os valores iniciais e finais de cada rota são iguais representando que o veículo inicia e termina a sua rota no mesmo depósito. Abaixo das rotas são apresentados o quanto da capacidade de cada veículo e de cada depósito foram utilizados.

Banco de Teste 1 (Barreto)

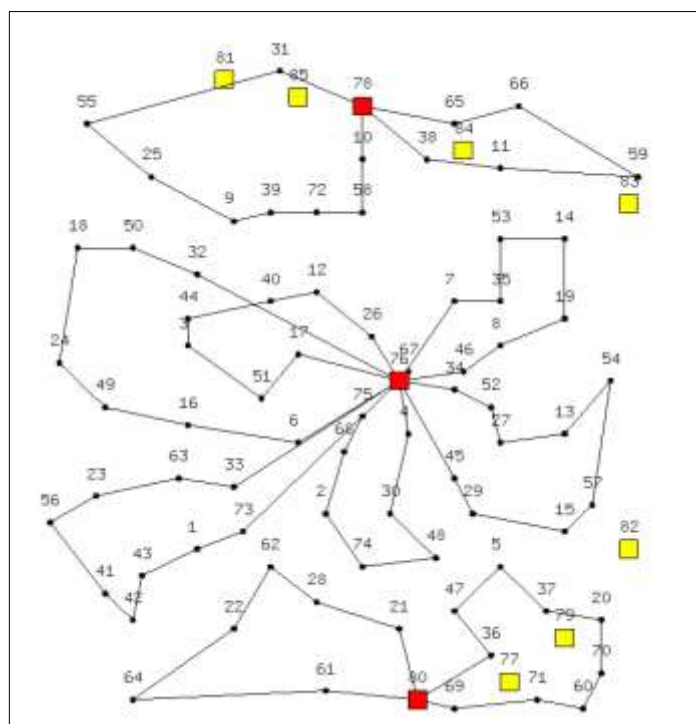


CH69-50x5

Custo Total: 565,61 (Solução Ótima)

Rotas													
Rota 1	52	4	42	19	40	41	13	25	18	52			
Rota 2	52	12	5	49	10	39	33	45	15	44	37	17	52
Rota 3	55	1	32	11	38	9	30	34	50	16	21	29	55
Rota 4	52	14	24	43	7	23	48	6	27	46	47	52	
Rota 5	55	22	8	26	31	28	3	36	35	20	2	55	

Capacidade Utilizada			
Rota 1	157	Deposito 51	0
Rota 2	158	Deposito 52	469
Rota 3	148	Deposito 53	0
Rota 4	154	Deposito 54	0
Rota 5	160	Deposito 55	308



CH75x10

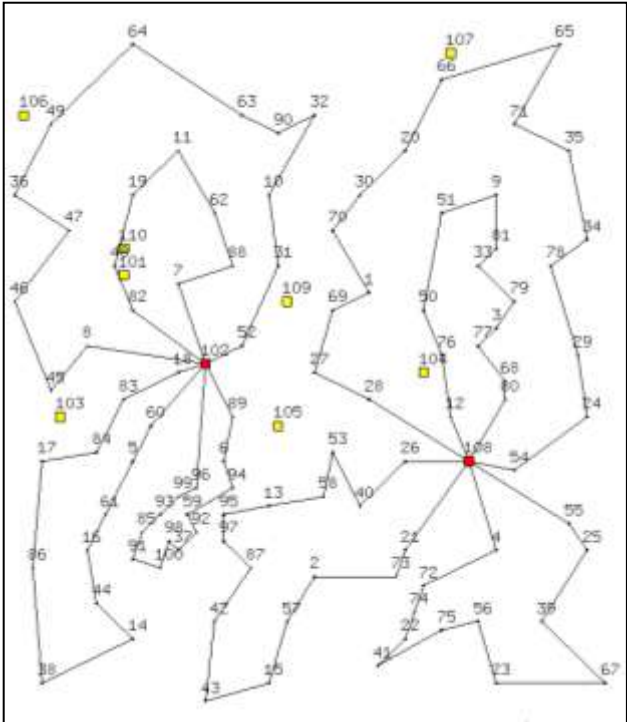
Custo Total: 845,93

Rotas

Rota 1	76	33	63	23	56	41	42	43	1	73	76
Rota 2	76	45	29	15	57	54	13	27	52	34	76
Rota 3	76	46	8	19	14	53	35	7	76		
Rota 4	76	26	12	40	44	3	51	17	76		
Rota 5	78	38	11	59	66	65	78				
Rota 6	80	69	71	60	70	20	37	5	47	36	80
Rota 7	76	4	30	48	74	2	68	75	76		
Rota 8	76	67	76								
Rota 9	80	61	64	22	62	28	21	80			
Rota 10	78	10	58	72	39	9	25	55	31	78	
Rota 11	76	32	50	18	24	49	16	6	76		

Capacidade Utilizada

Rota 1	138	Depósitos 76	841
Rota 2	139	Depósitos 77	0
Rota 3	136	Depósitos 78	270
Rota 4	127	Depósitos 79	0
Rota 5	131	Depósitos 80	253
Rota 6	123	Depósitos 81	0
Rota 7	138	Depósitos 82	0
Rota 8	30	Depósitos 83	0
Rota 9	130	Depósitos 84	0
Rota 10	139	Depósitos 85	0
Rota 11	133		



CH100x10

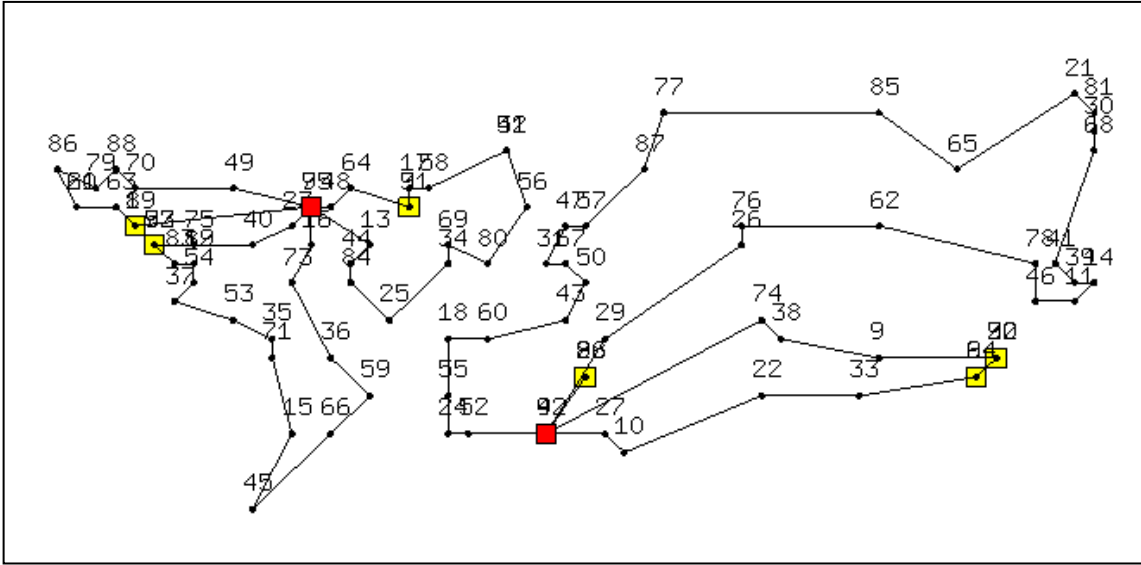
Custo Total: 833,40 (Solução Ótima)

Rotas

Rota 1	108	80	68	77	3	79	33	81	9	51	50	76	12	108					
Rota 2	102	96	99	93	85	91	100	98	37	92	59	94	6	89	102				
Rota 3	108	26	40	53	58	13	95	97	87	42	43	15	57	2	73	21	108		
Rota 4	102	18	83	84	17	86	38	14	44	16	61	5	60	102					
Rota 5	102	8	45	46	47	36	49	64	63	90	32	10	31	52	102				
Rota 6	102	7	88	62	11	19	48	82	102										
Rota 7	108	54	24	29	78	34	35	71	65	66	20	30	70	1	69	27	28	108	
Rota 8	108	4	72	74	22	41	75	56	23	67	39	25	55	108					

Capacidade Utilizada

Rota 1	200	Depósito 101	0
Rota 2	194	Depósito 102	675
Rota 3	193	Depósito 103	0
Rota 4	182	Depósito 104	0
Rota 5	185	Depósito 105	0
Rota 6	114	Depósito 106	0
Rota 7	198	Depósito 107	0
Rota 8	192	Depósito 108	783
		Depósito 109	0
		Depósito 110	0



Daskin 88x8

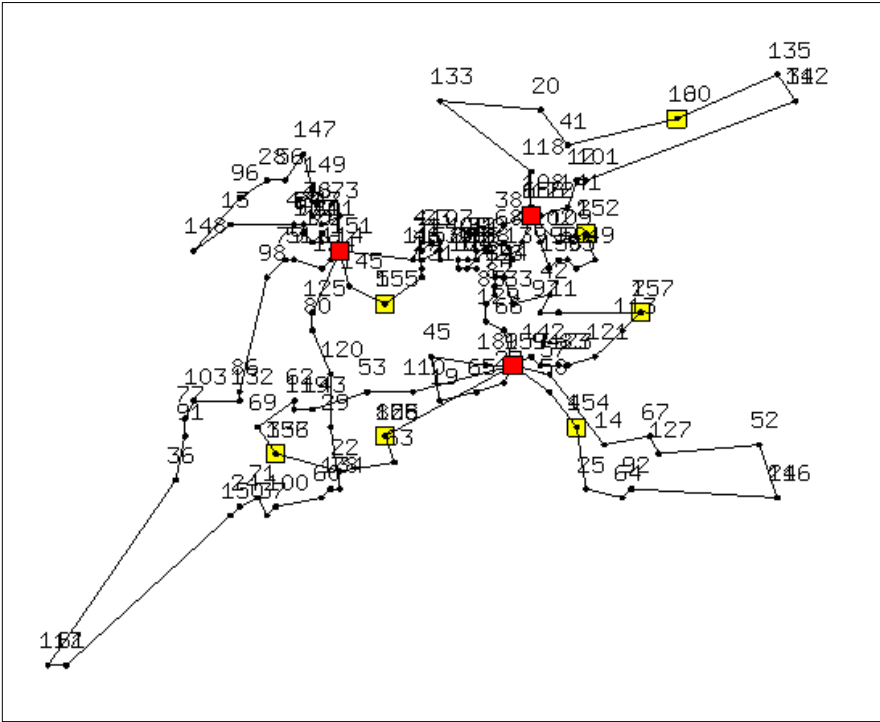
Custo Total: 355,78 (Solução Ótima)

Rotas

Rota 1	95	13	44	84	25	34	69	80	56	42	51	58	17	3	64	48	95					
Rota 2	92	74	38	9	2	32	6	33	22	10	27	92										
Rota 3	92	52	24	55	18	60	43	50	67	31	47	57	87	77	85	65	21	81	30	68		
Rota 3	41	39	14	11	46	78	62	76	26	29	92											
Rota 4	95	16	73	36	59	66	45	15	71	35	53	37	54	82	19	12	83	5	72	75		
Rota 4	40	23	7	95																		
Rota 5	95	49	70	88	79	86	61	20	63	1	95											
Rota 6	92	8	28	4	92																	

Capacidade Utilizada

Rota 1	7044247	Depósito 89	0
Rota 2	8771665	Depósito 90	0
Rota 3	8525752	Depósito 91	0
Rota 4	8721761	Depósito 92	20382466
Rota 5	8692097	Depósito 93	0
Rota 6	3085049	Depósito 94	0
		Depósito 95	24458105
		Depósito 96	0



Daskin 150x10

Custo Total: 43919,89

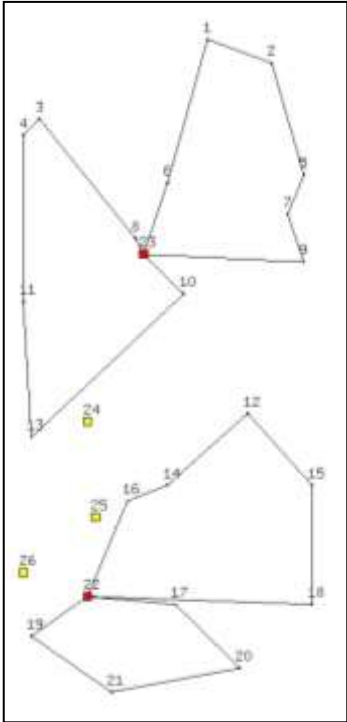
Rotas

Rota 1	156	40	139	68	32	128	59	16	72	104	88	106	35	39	116
Rota 1	30	81	55	90	38	156									
Rota 2	156	6	102	138	99	50	95	49	2	109	156				
Rota 3	159	110	53	143	119	62	69	136	3	22	63	105	8	126	159
Rota 4	159	57	14	67	127	52	21	146	92	64	25	4	58	159	
Rota 5	159	142	94	123	83	121	113	7	11	97	42	33	26	89	144
Rota 5	74	82	85	129	66	159									
Rota 6	151	114	124	51	79	98	86	132	103	77	91	36	117	61	150
Rota 6	24	71	37	100	60	134	78	29	120	80	125	151			
Rota 7	151	70	84	111	75	15	137	93	87	54	43	13	148	96	28
Rota 7	56	147	149	76	48	47	73	151							
Rota 8	151	115	44	31	23	107	130	140	46	17	131	5	145	151	
Rota 9	156	122	141	12	101	112	34	135	10	41	20	133	118	108	156
Rota 10	151	1	151												
Rota 11	159	18	45	19	65	27	9	159							

Capacidade Utilizada

Rota 1	7576360	Depósitos 151	29760335
Rota 2	6969974	Depósitos 152	0
Rota 3	7858578	Depósitos 153	0
Rota 4	7573312	Depósitos 154	0
Rota 5	7880882	Depósitos 155	0
Rota 6	7933003	Depósitos 156	20472668

Rota 7	7924028	Depósitos 157	0
Rota 8	6509495	Depósitos 158	0
Rota 9	5926334	Depósitos 159	27735382
Rota 10	7393809	Depósitos 160	0
Rota 11	4422610		

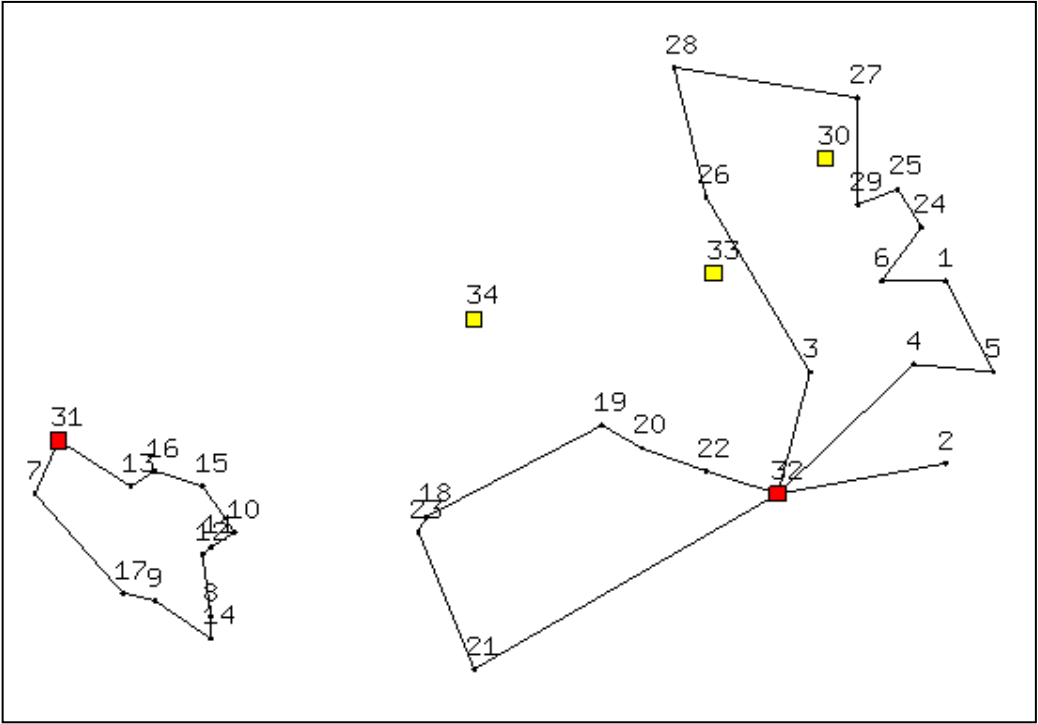


Gas 21x5

Custo Total: 424,90 (Solução Ótima)

Rotas								
Rota 1	23	8	3	4	11	13	10	23
Rota 2	22	17	20	21	19	22		
Rota 3	23	6	1	2	5	7	9	23
Rota 4	22	18	15	12	14	16	22	

Capacidade Utilizada			
Rota 1	5400	Depósito 22	11500
Rota 2	6000	Depósito 23	11000
Rota 3	5600	Depósito 24	0
Rota 4	5500	Depósito 25	0



Gas 29x5

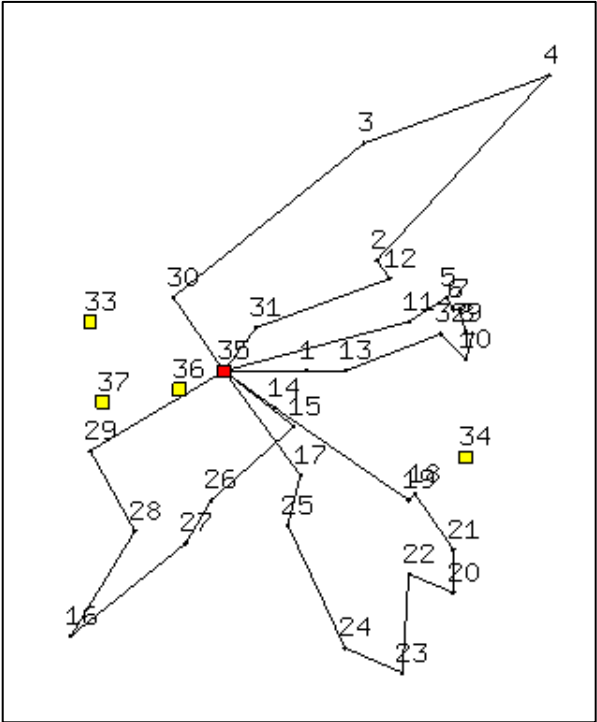
Custo Total: 512,10 (Solução Ótima)

Rotas

Rota 1	32	3	26	28	27	29	25	24	6	1	5	4	32
Rota 2	32	2	32										
Rota 3	32	22	20	19	18	23	21	32					
Rota 4	31	13	16	15	10	11	12	8	14	9	17	7	31

Capacidade Utilizada

Rota 1	3725	Depósito 30	0
Rota 2	3100	Depósito 31	3175
Rota 3	2750	Depósito 32	9575
Rota 4	3175	Depósito 33	0
		Depósito 34	0



Gas 32x5A

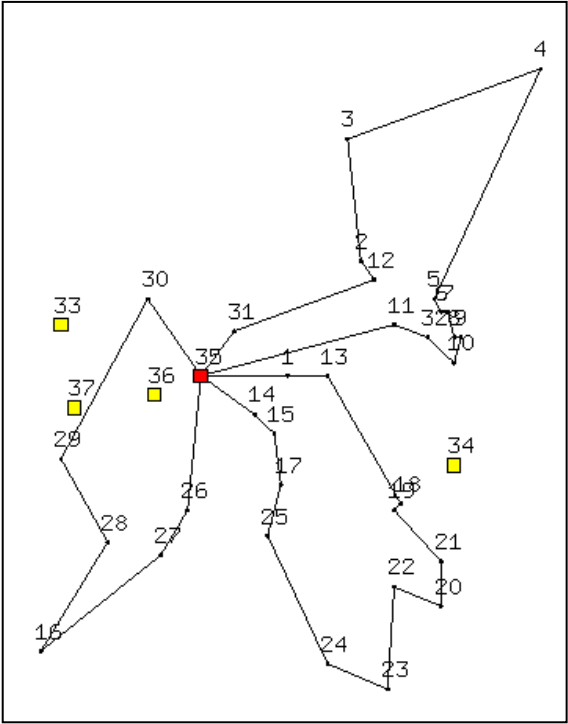
Custo Total: 562,22 (Solução Ótima)

Rotas

Rota 1	35	29	28	16	27	26	15	35				
Rota 2	35	11	5	6	7	8	9	10	32	13	1	35
Rota 3	35	14	19	18	21	20	22	23	24	25	17	35
Rota 4	35	30	3	4	2	12	31	35				

Capacidade Utilizada

Rota 1	7250	Depósito 33	0
Rota 2	7920	Depósito 34	0
Rota 3	7850	Depósito 35	29370
Rota 4	6350	Depósito 36	0
		Depósito 37	0



Gas 32x5B

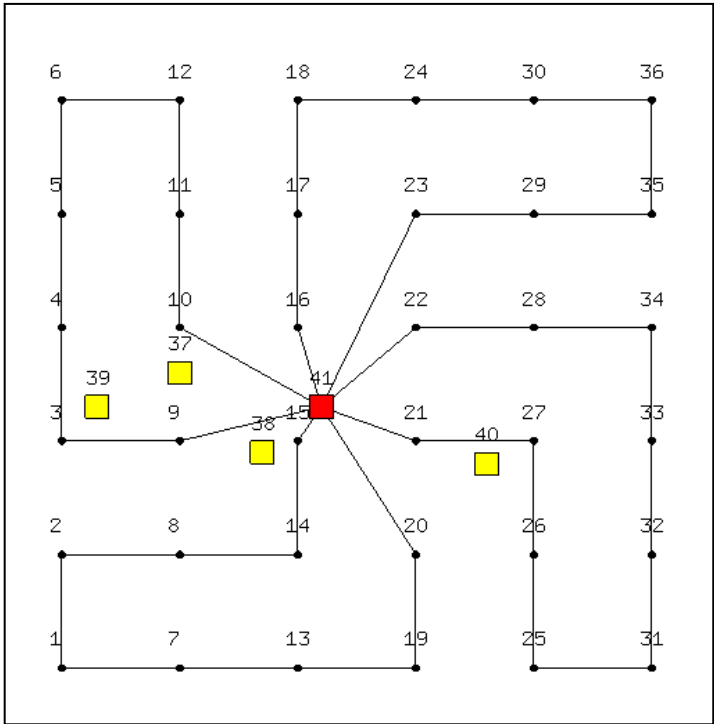
Custo Total: 504,30 (Solução Ótima)

Rotas

Rota 1	35	1	13	18	19	21	20	22	23	24	25	17	15	14	35
Rota 2	35	26	27	16	28	29	30	35							
Rota 3	35	11	32	10	9	8	7	6	5	4	3	2	12	31	35

Capacidade Utilizada

Rota 1	9250	Depósitos 33	0
Rota 2	9300	Depósitos 34	0
Rota 3	10820	Depósitos 35	29370
		Depósitos 36	0
		Depósitos 37	0



Gas 36x5

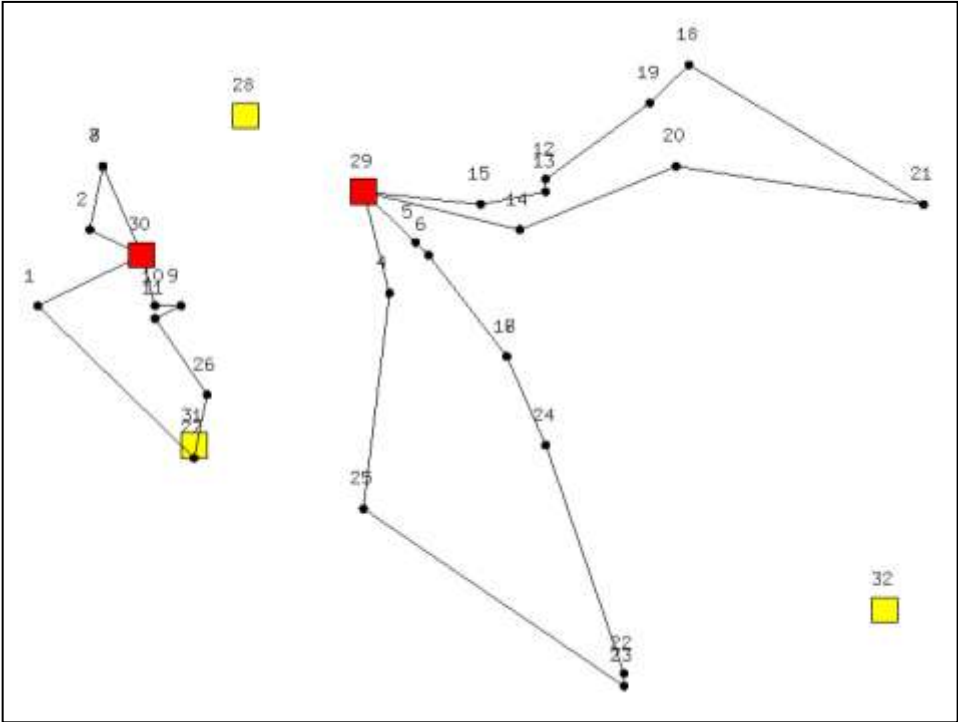
Custo Total: 460,40 (Solução Ótima)

Rotas

Rota 1	41	21	27	33	34	35	36	30	29	28	22	41
Rota 2	41	15	14	13	19	25	31	32	26	20	41	
Rota 3	41	23	24	18	12	6	5	11	17	16	41	
Rota 4	41	9	8	7	1	2	3	4	10	41		

Capacidade Utilizada

Rota 1	250	Depósitos 37	0
Rota 2	225	Depósitos 38	0
Rota 3	225	Depósitos 39	0
Rota 4	200	Depósitos 40	0
		Depósitos 41	900

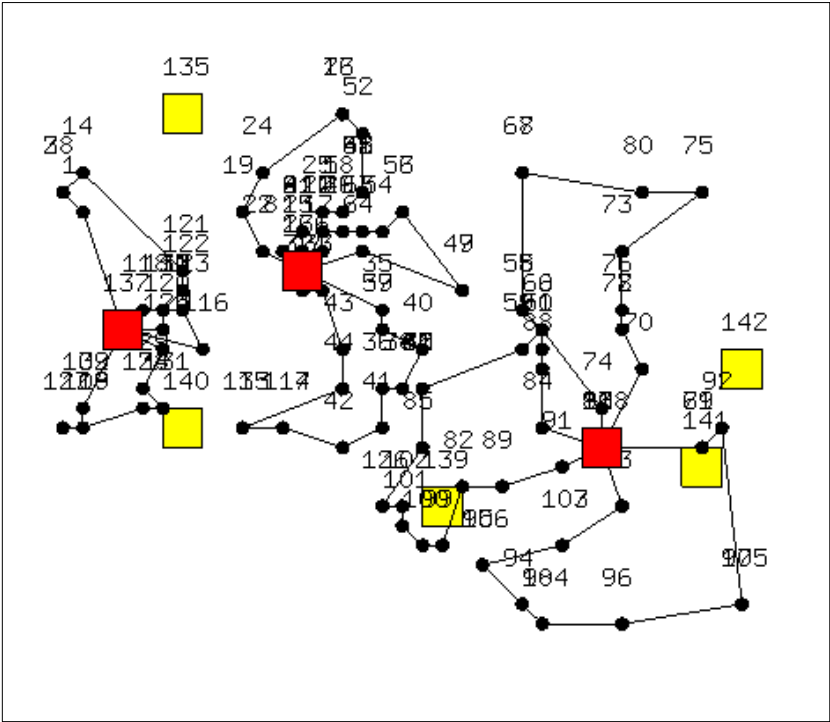


Min 27x5

Custo Total: 3062,00 (Solução Ótima)

Rotas										
Rota 1	30	1	27	26	11	9	10	30		
Rota 2	29	14	20	21	18	19	12	13	15	29
Rota 3	30	3	8	7	2	30				
Rota 4	29	5	6	17	16	24	22	23	25	4 29

Capacidade Utilizada			
Rota 1	2300	Depósitos 28	0
Rota 2	2280	Depósitos 29	4540
Rota 3	1570	Depósitos 30	3870
Rota 4	2260	Depósitos 31	0
		Depósitos 32	0



Min 134x8

Custo Total: 5719,79

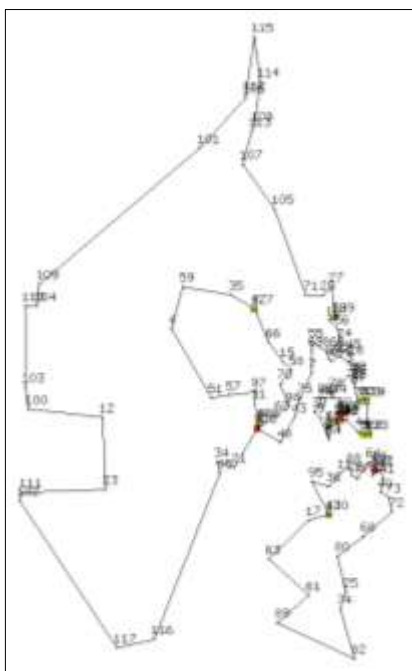
Rotas

Rota 1	136	12	27	11	10	25	5	18	48	62	65	51	52	13	16
Rota 1	7	24	19	22	136										
Rota 2	136	26	8	21	9	4	6	23	15	136					
Rota 3	136	64	47	49	56	53	54	63	29	46	20	17	136		
Rota 4	137	1	2	28	3	14	121	122	113	130	120	137			
Rota 5	138	93	83	107	103	95	106	94	104	98	96	105	97	92	71
Rota 5	69	79	81	138											
Rota 6	137	132	109	119	127	108	128	134	131	129	124	125	137		
Rota 7	137	118	111	110	112	123	116	137							
Rota 8	136	35	57	39	40	38	34	36	41	42	114	117	115	133	44
Rota 8	43	33	32	37	136										
Rota 9	138	90	70	78	72	77	76	73	75	80	68	67	55	58	60
Rota 9	74	138													
Rota 10	138	91	89	82	99	100	101	102	126	85	30	45	31	87	59
Rota 10	66	61	50	88	84	138									
Rota 11	138	86	138												

Capacidade Utilizada

Rota 1	835	Depósitos 135	0
Rota 2	431	Depósitos 136	2894
Rota 3	791	Depósitos 137	2478
Rota 4	821	Depósitos 138	2539
Rota 5	829	Depósitos 139	0
Rota 6	811	Depósitos 140	0

Rota 7	846	Depósitos 141	0
Rota 8	837	Depósitos 142	0
Rota 9	841		
Rota 10	841		
Rota 11	28		



Or-117x14

Custo Total: 12296,93

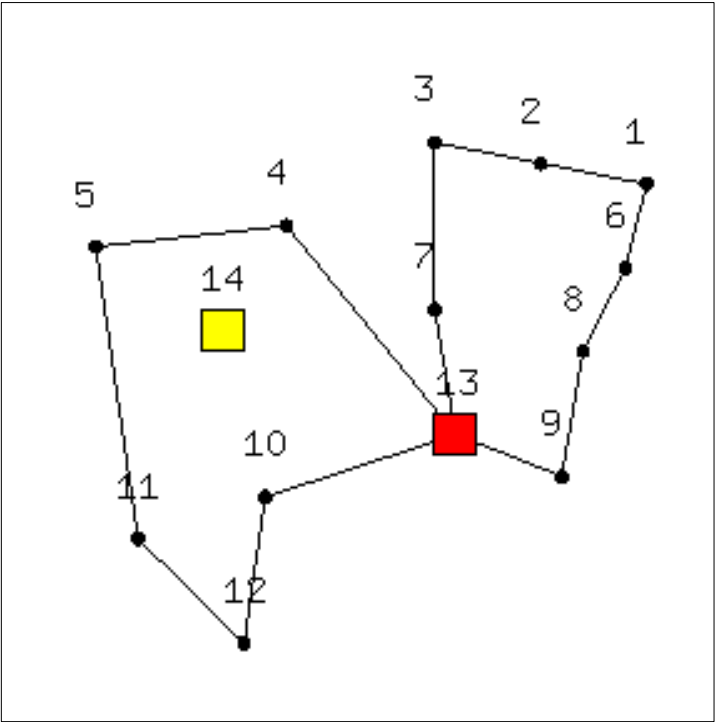
Rotas

Rota 1	118	22	53	76	13	54	9	30	29	96	60	84	78	38	83
Rota 1	19	16	32	6	67	33	93	1	2	52	44	81	39	118	
Rota 2	120	21	34	85	116	117	102	111	23	12	100	103	110	104	109
Rota 2	101	106	112	115	114	108	113	107	105	71	20	27	79	10	56
Rota 2	24	8	45	28	5	65	49	87	86	55	69	7	75	43	48
Rota 2	120														
Rota 3	120	46	31	97	57	51	4	59	35	47	66	15	58	70	98
Rota 3	62	92	120												
Rota 4	118	18	3	50	94	91	118								
Rota 5	121	99	37	41	40	73	72	68	80	25	74	82	89	61	63
Rota 5	17	42	95	36	11	88	26	77	64	14	90	121			

Capacidade Utilizada

Rota 1	143696	Depósitos 118	262499
Rota 2	146648	Depósitos 119	0
Rota 3	106803	Depósitos 120	253451
Rota 4	118803	Depósitos 121	129579
Rota 5	129579	Depósitos 122	0
		Depósitos 123	0
		Depósitos 124	0
		Depósitos 125	0
		Depósitos 126	0
		Depósitos 127	0
		Depósitos 128	0
		Depósitos 129	0

Depósitos 130	0
Depósitos 131	0

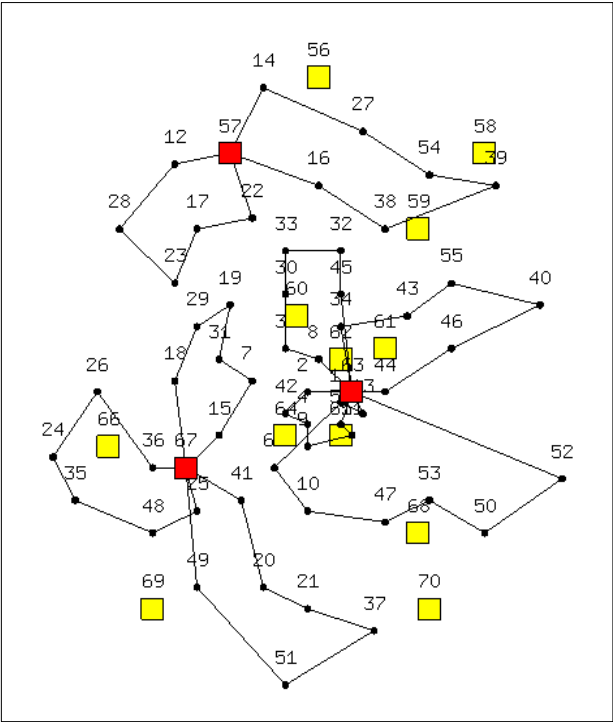


Perl 12x2

Custo Total: 203,98 (Solução Ótima)

Rotas									
Rota 1	13	4	5	11	12	10	13		
Rota 2	13	7	3	2	1	6	8	9	13

Capacidade Utilizada			
Rota 1	100	Depósitos 13	240
Rota 2	140	Depósitos 14	0



Perl 55x15

Custo Total: 1112,05 (Solução Ótima)

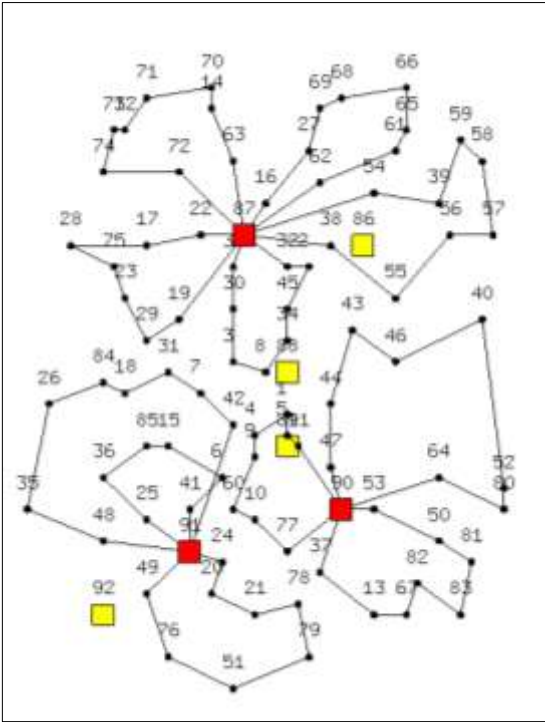
Rotas

Rota 1	63	2	42	4	9	11	5	63
Rota 2	67	18	29	19	31	7	15	67
Rota 3	63	52	50	53	47	10	6	63
Rota 4	67	36	26	24	35	48	25	67
Rota 5	63	8	3	30	33	32	45	63
Rota 6	57	22	17	23	28	12	57	
Rota 7	57	16	38	39	54	27	14	57
Rota 8	67	41	20	21	37	51	49	67
Rota 9	63	34	43	55	40	46	44	63
Rota 10	63	13	1	63				

Capacidade Utilizada

Rota 1	120	Depósitos 56	0
Rota 2	120	Depósitos 57	220
Rota 3	120	Depósitos 58	0
Rota 4	120	Depósitos 59	0
Rota 5	120	Depósitos 60	0
Rota 6	100	Depósitos 61	0
Rota 7	120	Depósitos 62	0
Rota 8	120	Depósitos 63	520
Rota 9	120	Depósitos 64	0
Rota 10	40	Depósitos 65	0
		Depósitos 66	0

	Depósitos 67	360
	Depósitos 68	0
	Depósitos 69	0
	Depósitos 70	0



Perl 85x7

Custo Total: 1622,49

Rotas

Rota 1	87	32	2	45	34	8	3	30	33	87
Rota 2	91	24	20	21	78	79	51	76	49	91
Rota 3	91	25	36	85	15	6	41	91		
Rota 4	90	77	10	60	9	4	1	5	11	90
Rota 5	87	16	27	69	68	66	65	61	62	87
Rota 6	87	72	74	73	12	71	70	14	63	87
Rota 7	90	47	44	43	46	40	52	80	64	90
Rota 8	91	42	7	31	18	84	26	35	48	91
Rota 9	87	19	29	23	75	28	17	22	87	
Rota 10	87	38	55	56	57	58	59	39	54	87
Rota 11	90	37	13	67	82	83	81	50	53	90

Capacidade Utilizada

Rota 1	160	Depósitos 86	0
Rota 2	160	Depósitos 87	780
Rota 3	120	Depósitos 88	0
Rota 4	160	Depósitos 89	0
Rota 5	160	Depósitos 90	480
Rota 6	160	Depósitos 91	440
Rota 7	160	Depósitos 92	0
Rota 8	160		
Rota 9	140		
Rota 10	160		
Rota 11	160		

Rotas															
Rota 1	319	95	108	59	31	71	70	69	4	139	50	46	104	112	158
Rota 1	198	142	144	207	219	154	148	151	197	199	164	166	163	165	205
Rota 1	146	222	221	220	177	145	185	190	189	171	319				
Rota 2	320	21	140	141	2	3	289	252	290	243	304	303	291	292	233
Rota 2	13	127	128	131	138	135	129	134	133	136	132	137	130	79	320
Rota 3	319	35	96	92	94	68	67	116	121	118	119	120	117	91	125
Rota 3	76	78	77	16	126	9	20	82	109	110	80	57	44	48	1
Rota 3	52	40	88	89	24	7	30	29	28	12	55	15	97	98	86
Rota 3	54	85	17	14	231	274	318	267	8	19	167	152	194	100	36
Rota 3	32	204	203	156	206	155	149	160	319	32					
Rota 4	322	298	244	297	271	270	308	302	293	305	153	147	216	214	215
Rota 4	213	212	217	218	143	201	200	202	168	169	254	261	295	238	237
Rota 4	241	322													
Rota 5	322	316	313	314	311	317	315	312	310	322					
Rota 6	321	232	278	246	255	262	251	249	250	235	63	62	83	5	41
Rota 6	61	114	101	107	66	65	56	87	102	124	45	43	42	103	49
Rota 6	38	25	75	27	74	10	93	113	99	90	47	51	6	106	37
Rota 6	122	72	111	22	33	64	115	123	26	34	226	73	105	81	84
Rota 6	23	11	39	18	225	224	223	253	309	239	248	242	258	265	260
Rota 6	229	228	263	245	264	236	230	240	247	321					
Rota 7	321	286	276	277	275	283	279	282	285	281	268	272	301	300	299
Rota 7	294	306	234	287	256	257	266	259	227	307	288	296	273	269	284
Rota 7	280	321													

Rota 8	319	192	176	172	181	173	191	186	184	182	174	175	179	159	211
Rota 8	58	60	210	209	150	208	157	161	196	187	162	193	170	188	183
Rota 8	180	178	195	319											

Capacidade Utilizada

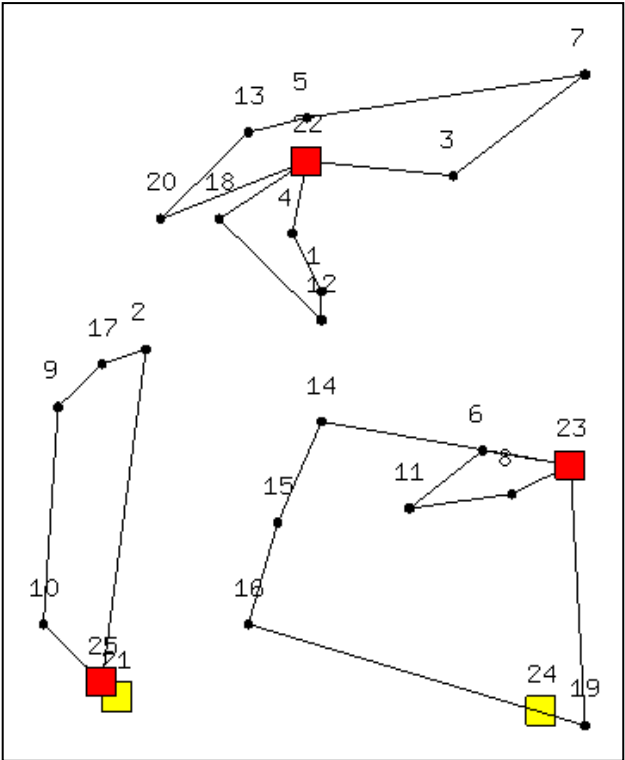
Rota 1	2117300	Depósitos 319	7064900
Rota 2	2195775	Depósitos 320	2195775
Rota 3	2470200	Depósitos 321	4788800
Rota 4	2491100	Depósitos 322	4760050
Rota 5	2268950		
Rota 6	2343475		
Rota 7	2445325		
Rota 8	2477400		

Rota 17	320	130	320												
Rota 18	319	180	145	208	157	150	209	210	60	319					
Rota 19	320	129	320												
Rota 20	319	195	183	177	220	221	222	146	205	165	163	166	164	199	197
Rota 20	151	148	154	219	207	144	142	198	158	112	104	46	50	4	69
Rota 20	70	31	71	319											
Rota 21	319	171	189	188	190	185	178	319							
Rota 22	319	174	182	184	186	191	173	181	172	176	192	319			
Rota 23	322	234	306	294	299	227	259	266	257	256	287	316	322		
Rota 24	319	149	156	212	213	206	155	160	319						
Rota 25	320	89	88	1	110	109	80	82	74	27	75	25	38	49	103
Rota 25	42	43	45	124	102	87	22	111	72	122	37	106	6	51	47
Rota 25	90	99	113	93	10	20	9	126	16	91	125	78	77	76	57
Rota 25	44	48	320												
Rota 26	319	95	319												

Capacidade Utilizada

Rota 1	784475	Depósitos 319	6740575
Rota 2	775700	Depósitos 320	4550825
Rota 3	696050	Depósitos 321	3062950
Rota 4	771925	Depósitos 322	4455175
Rota 5	791725		
Rota 6	746625		
Rota 7	780125		
Rota 8	768400		
Rota 9	773325		
Rota 10	798625		
Rota 11	645250		
Rota 12	597550		
Rota 13	769500		
Rota 14	722850		
Rota 15	780125		
Rota 16	707925		
Rota 17	478300		
Rota 18	799500		
Rota 19	479525		
Rota 20	791575		
Rota 21	592550		
Rota 22	762800		
Rota 23	797400		
Rota 24	792225		
Rota 25	755525		
Rota 26	649950		

Banco de Teste 2 (Prins)

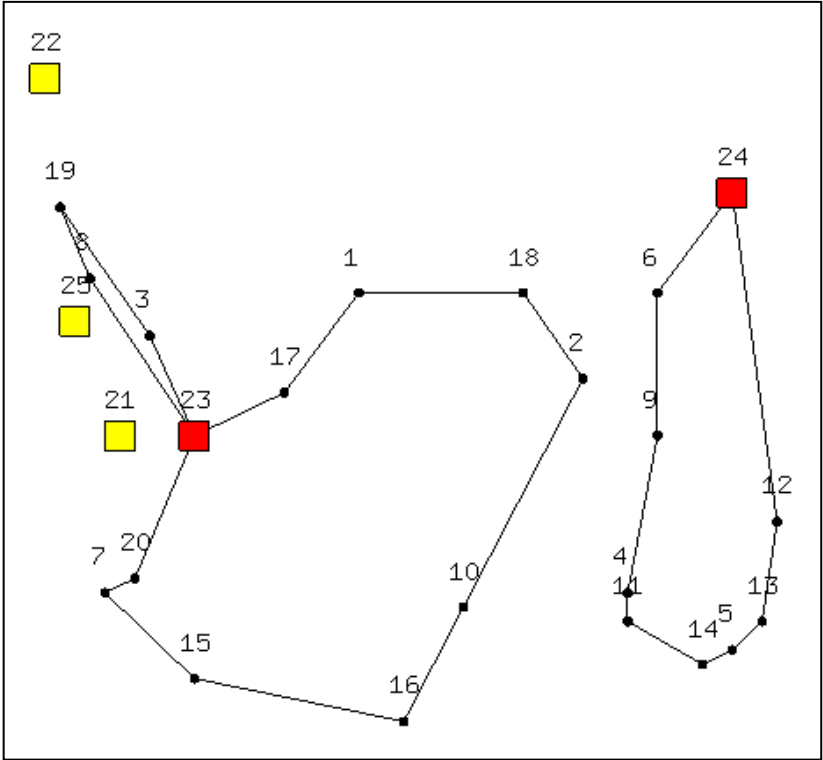


Coord 20-5-1

Custo Total: 54793,00 (Solução Ótima)

Rotas							
Rota 1	22	4	1	12	18	22	
Rota 2	22	20	13	5	7	3	22
Rota 3	23	19	16	15	14	23	
Rota 4	23	6	11	8	23		
Rota 5	25	2	17	9	10	25	

Capacidade Utilizada			
Rota 1	69	Depósitos 21	0
Rota 2	69	Depósitos 22	138
Rota 3	60	Depósitos 23	107
Rota 4	47	Depósitos 24	0
Rota 5	70	Depósitos 25	70

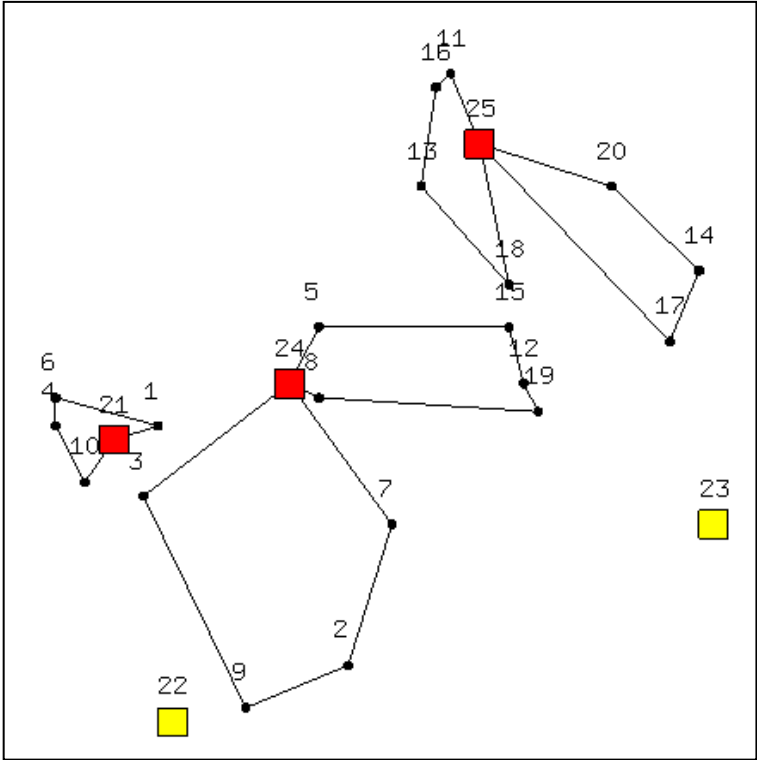


Coord 20-5-1b

Custo Total: 39104,00 (Solução Ótima)

Rotas											
Rota 1	23	8	19	3	23						
Rota 2	23	20	7	15	16	10	2	18	1	17	23
Rota 3	24	6	9	4	11	14	5	13	12	24	

Capacidade Utilizada			
Rota 1	50	Depósitos 21	0
Rota 2	140	Depósitos 22	0
Rota 3	118	Depósitos 23	190
		Depósitos 24	118
		Depósitos 25	0



Coord 20-5-2

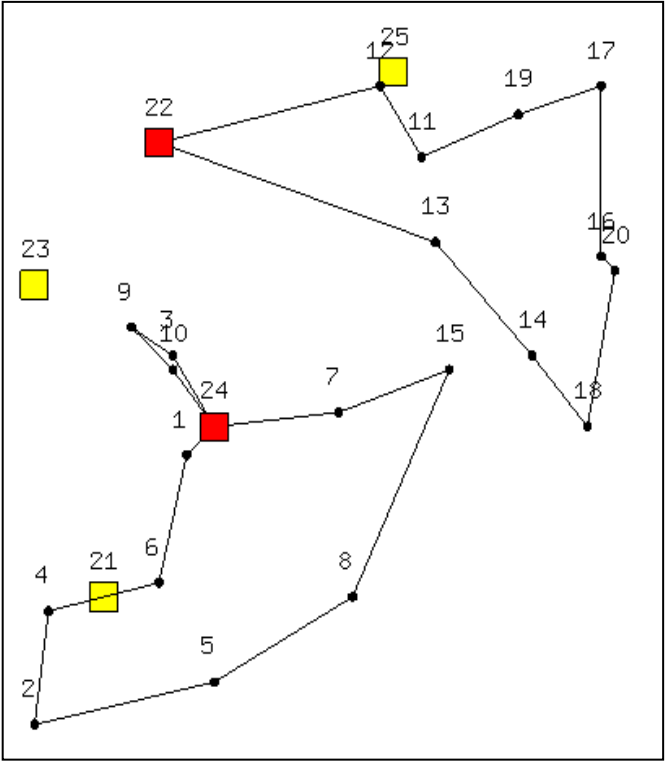
Custo Total: 48908,00 (Solução Ótima)

Rotas

Rota 1	25	11	16	13	18	25	
Rota 2	21	10	4	6	1	21	
Rota 3	24	8	19	12	15	5	24
Rota 4	25	20	14	17	25		
Rota 5	24	3	9	2	7	24	

Capacidade Utilizada

Rota 1	60	Depósitos 21	64
Rota 2	64	Depósitos 22	0
Rota 3	67	Depósitos 23	0
Rota 4	53	Depósitos 24	133
Rota 5	66	Depósitos 25	113

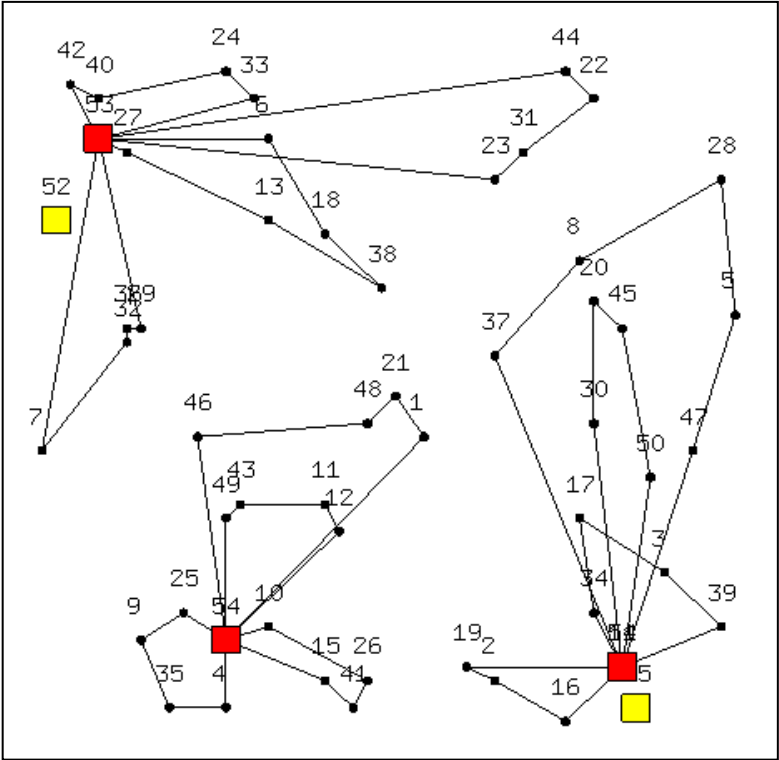


Coord 20-5-2b

Custo Total: 37542,00 (Solução Ótima)

Rotas											
Rota 1	22	12	11	19	17	16	20	18	14	13	22
Rota 2	24	3	9	10	24						
Rota 3	24	7	15	8	5	2	4	6	1	24	

Capacidade Utilizada			
Rota 1	144	Depósitos 21	0
Rota 2	41	Depósitos 22	144
Rota 3	117	Depósitos 23	0
		Depósitos 24	158
		Depósitos 25	0



Coord 50-5-1

Custo Total: 90111,00 (Solução Ótima)

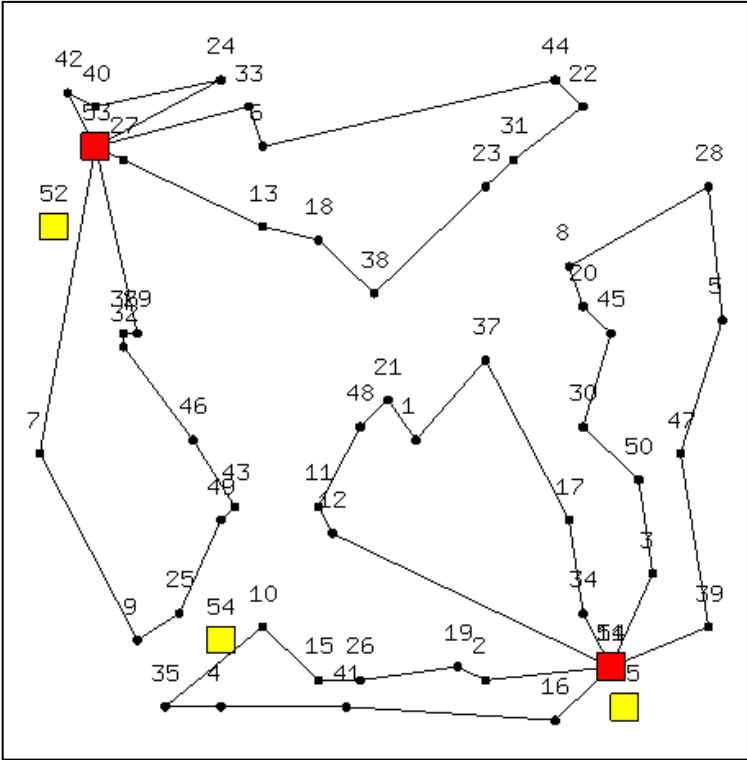
Rotas

Rota 1	53	27	13	38	18	6	53
Rota 2	54	49	43	11	12	54	
Rota 3	53	44	22	31	23	53	
Rota 4	51	37	8	28	5	47	51
Rota 5	54	46	48	21	1	54	
Rota 6	53	33	24	40	42	53	
Rota 7	54	15	41	26	10	54	
Rota 8	51	34	17	3	39	51	
Rota 9	54	4	35	9	25	54	
Rota 10	51	50	45	20	30	51	
Rota 11	53	7	32	36	29	53	
Rota 12	51	14	19	2	16	51	

Capacidade Utilizada

Rota 1	68	Depósitos 51	244
Rota 2	64	Depósitos 52	0
Rota 3	62	Depósitos 53	256
Rota 4	68	Depósitos 54	256
Rota 5	69	Depósitos 55	0
Rota 6	58		
Rota 7	63		
Rota 8	60		
Rota 9	60		

Rota 10	60
Rota 11	68
Rota 12	56



Coord 50-5-1b

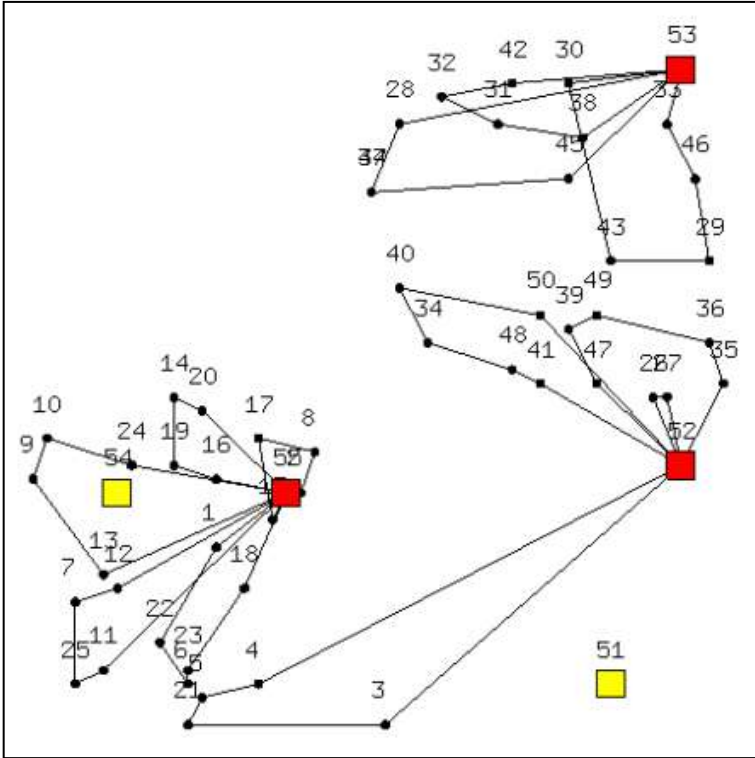
Custo Total: 63242,00 (Solução Ótima)

Rotas

Rota 1	51	2	19	26	15	10	35	4	41	16	51	
Rota 2	53	7	9	25	49	43	46	32	36	29	53	
Rota 3	51	14	12	11	48	21	1	37	17	34	51	
Rota 4	51	39	47	5	28	8	20	45	30	50	3	51
Rota 5	53	33	6	44	22	31	23	38	18	13	27	53
Rota 6	53	24	40	42	53							

Capacidade Utilizada

Rota 1	134	Depósitos 51	420
Rota 2	148	Depósitos 52	0
Rota 3	136	Depósitos 53	336
Rota 4	150	Depósitos 54	0
Rota 5	149	Depósitos 55	0
Rota 6	39		

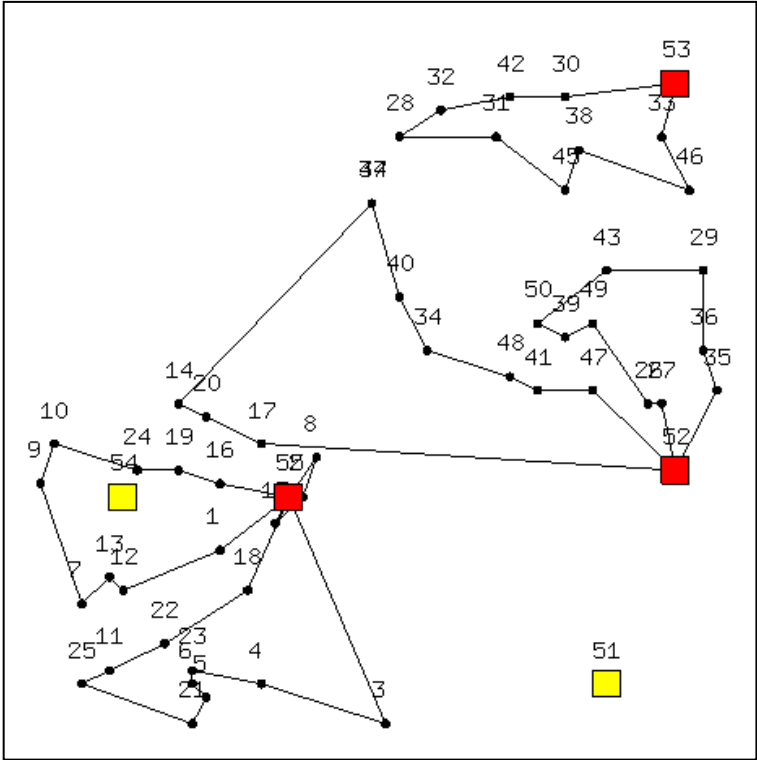


Coord 50-5-2

Custo Total: 88298,00 (Solução Ótima)

Rotas							
Rota 1	53	33	46	29	43	30	53
Rota 2	55	18	23	6	22	1	55
Rota 3	52	35	36	49	39	47	52
Rota 4	52	50	40	34	48	41	52
Rota 5	53	38	31	32	42	53	
Rota 6	52	4	5	21	3	52	
Rota 7	52	27	26	52			
Rota 8	55	20	14	19	16	55	
Rota 9	55	12	7	25	11	55	
Rota 10	53	28	44	37	45	53	
Rota 11	55	15	17	8	2	55	
Rota 12	55	24	10	9	13	55	

Capacidade Utilizada			
Rota 1	70	Depósitos 51	0
Rota 2	68	Depósitos 52	240
Rota 3	70	Depósitos 53	209
Rota 4	69	Depósitos 54	0
Rota 5	70	Depósitos 55	326
Rota 6	64		
Rota 7	37		
Rota 8	62		
Rota 9	64		
Rota 10	69		
Rota 11	69		
Rota 12	63		

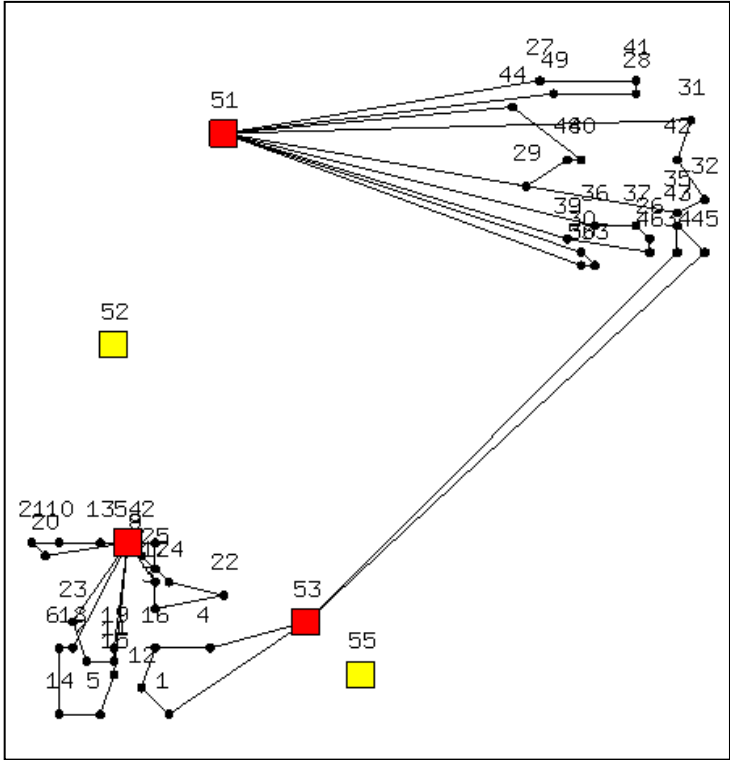


Coord 50-5-2b

Custo Total: 67308,00 (Solução Ótima)

Rotas												
Rota 1	55	3	4	23	6	5	21	25	11	22	18	55
Rota 2	52	17	20	14	44	37	40	34	48	41	47	52
Rota 3	53	30	42	32	28	31	45	38	46	33	53	
Rota 4	55	16	19	24	10	9	7	13	12	1	55	
Rota 5	55	15	2	8	55							
Rota 6	52	35	36	29	43	50	39	49	26	27	52	

Capacidade Utilizada			
Rota 1	148	Depósitos 51	0
Rota 2	150	Depósitos 52	295
Rota 3	140	Depósitos 53	140
Rota 4	142	Depósitos 54	0
Rota 5	50	Depósitos 55	340
Rota 6	145		



Coord 50-5-2bis

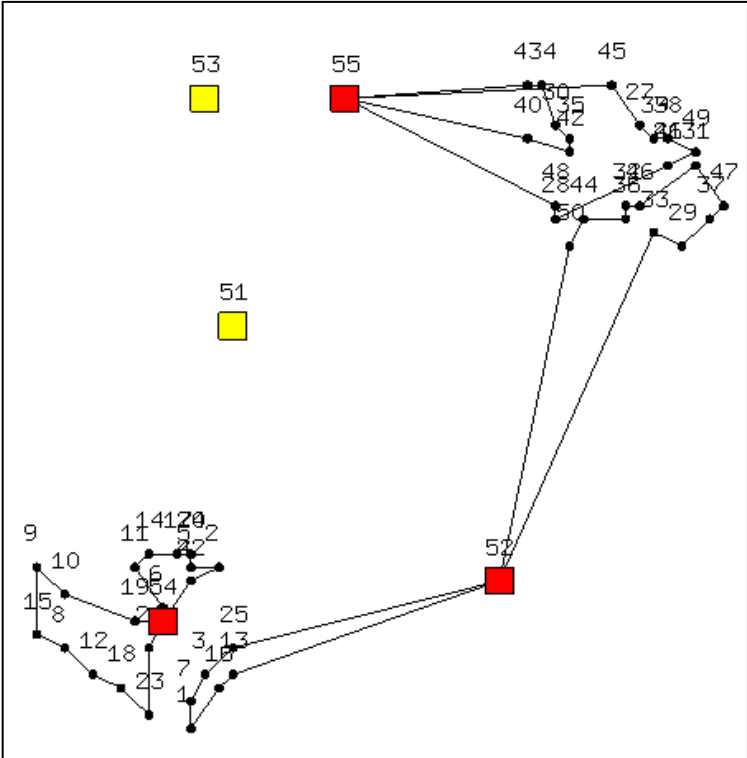
Custo Total: 84055,00 (Solução Ótima)

Rotas

Rota 1	54	18	6	14	5	15	54
Rota 2	51	30	33	38	50	51	
Rota 3	54	24	22	3	17	54	
Rota 4	54	13	10	21	20	54	
Rota 5	53	45	47	43	34	53	
Rota 6	54	8	9	25	2	54	
Rota 7	54	23	7	11	19	54	
Rota 8	53	1	12	16	4	53	
Rota 9	51	39	46	26	37	36	51
Rota 10	51	49	28	41	27	51	
Rota 11	51	35	32	42	31	51	
Rota 12	51	29	48	40	44	51	

Capacidade Utilizada

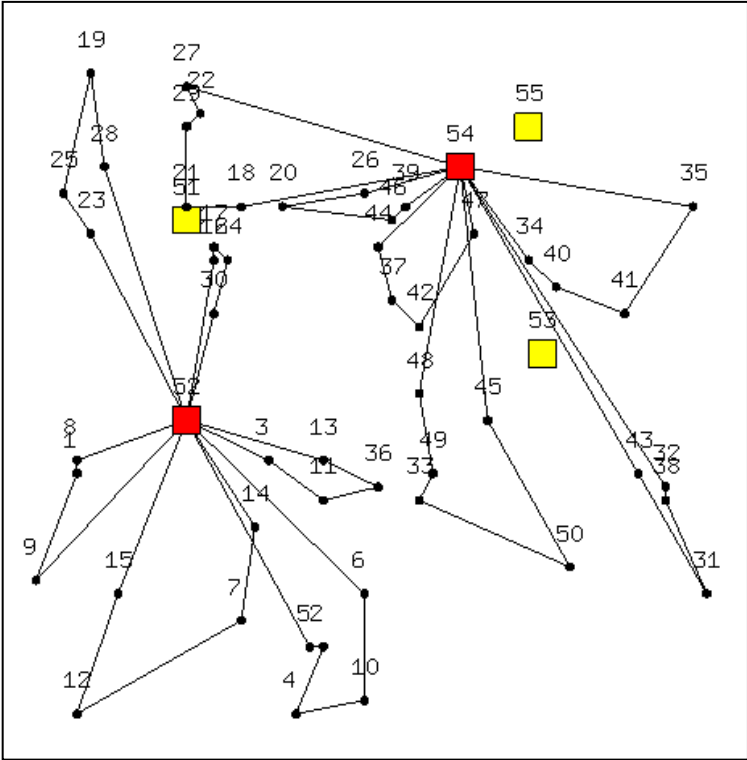
Rota 1	70	Depósitos 51	332
Rota 2	64	Depósitos 52	0
Rota 3	63	Depósitos 53	122
Rota 4	63	Depósitos 54	315
Rota 5	66	Depósitos 55	0
Rota 6	53		
Rota 7	66		
Rota 8	56		
Rota 9	70		
Rota 10	67		
Rota 11	63		
Rota 12	68		



Custo Total: 57883,00 (Solução Ótima)

Rotas												
Rota 1	54	6	11	14	17	20	24	5	2	4	22	54
Rota 2	52	33	29	37	47	31	46	32	36	44	50	52
Rota 3	54	19	10	9	15	8	12	18	23	21	54	
Rota 4	52	25	3	7	1	16	13	52				
Rota 5	55	48	28	41	26	49	38	39	27	45	55	
Rota 6	55	40	42	35	30	34	43	55				

Capacidade Utilizada			
Rota 1	145	Depósitos 51	0
Rota 2	150	Depósitos 52	250
Rota 3	141	Depósitos 53	0
Rota 4	100	Depósitos 54	286
Rota 5	149	Depósitos 55	247
Rota 6	98		



Coord 50-5-3

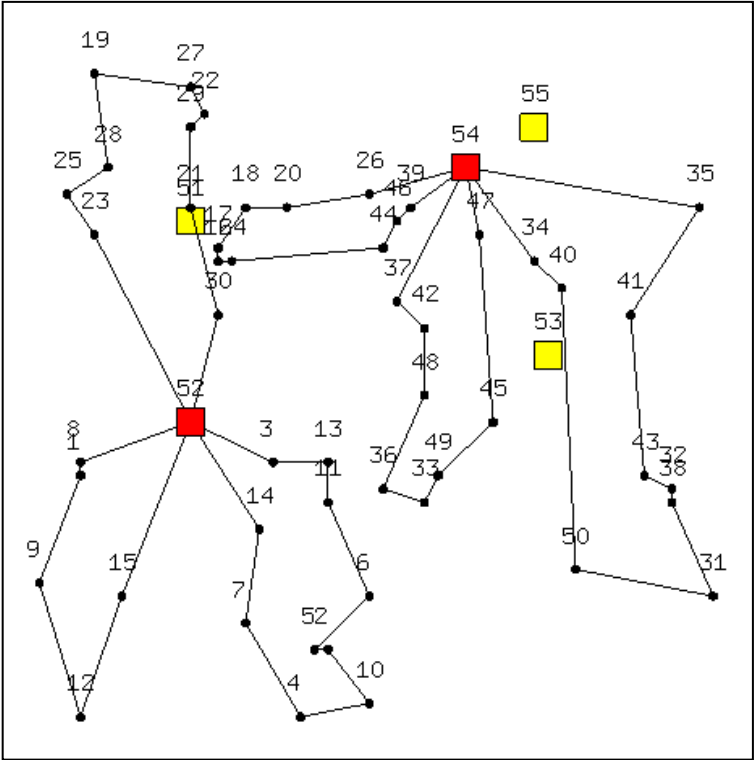
Custo Total: 86203,00 (Solução Ótima)

Rotas

Rota 1	54	27	22	29	21	18	54
Rota 2	54	43	31	38	32	54	
Rota 3	52	23	25	19	28	52	
Rota 4	52	16	17	24	30	52	
Rota 5	54	26	20	46	39	54	
Rota 6	52	5	2	4	10	6	52
Rota 7	52	8	1	9	52		
Rota 8	52	15	12	7	14	52	
Rota 9	54	45	50	33	49	48	54
Rota 10	52	3	11	36	13	52	
Rota 11	54	35	41	40	34	54	
Rota 12	54	44	37	42	47	54	

Capacidade Utilizada

Rota 1	69	Depósitos 101	0
Rota 2	63	Depósitos 102	372
Rota 3	66	Depósitos 103	0
Rota 4	61	Depósitos 104	389
Rota 5	69	Depósitos 105	0
Rota 6	65	Depósitos 106	
Rota 7	49	Depósitos 107	
Rota 8	69	Depósitos 108	
Rota 9	66	Depósitos 109	
Rota 10	62	Depósitos 110	
Rota 11	64		
Rota 12	58		



Coord 50-5-3b

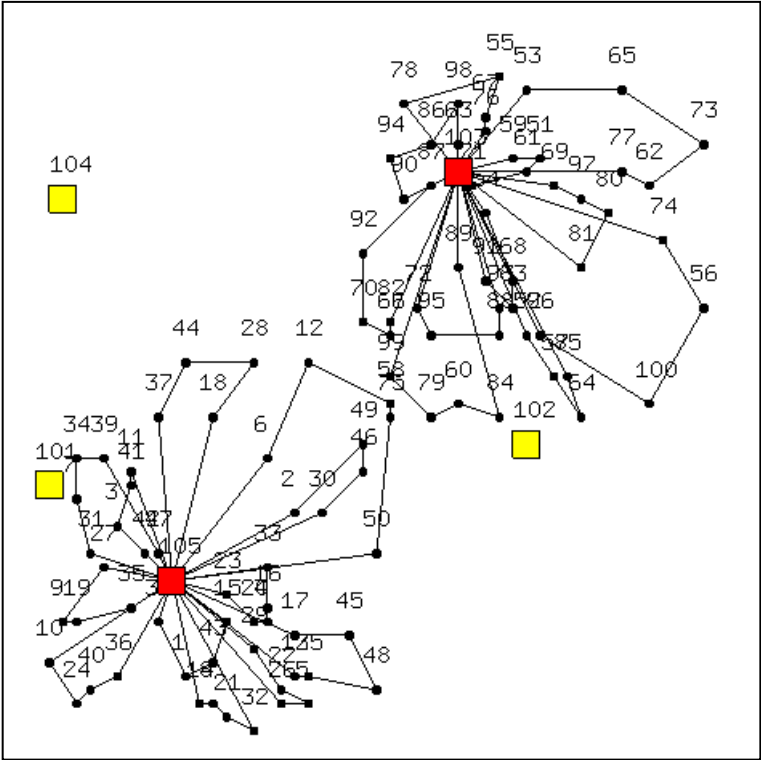
Custo Total: 86203,00 (Solução Ótima)

Rotas

Rota 1	54	34	40	50	31	38	32	43	41	35	54	52
Rota 2	54	39	46	44	24	16	17	18	20	26	54	
Rota 3	52	14	7	4	10	2	5	6	11	13	3	
Rota 4	52	23	25	28	19	27	22	29	21	30	52	
Rota 5	52	15	12	9	1	8	52					
Rota 6	54	47	45	49	33	36	48	42	37	54		

Capacidade Utilizada

Rota 1	142	Depósitos 101	0
Rota 2	148	Depósitos 102	361
Rota 3	141	Depósitos 103	0
Rota 4	132	Depósitos 104	400
Rota 5	88	Depósitos 105	0
Rota 6	110	Depósitos 106	

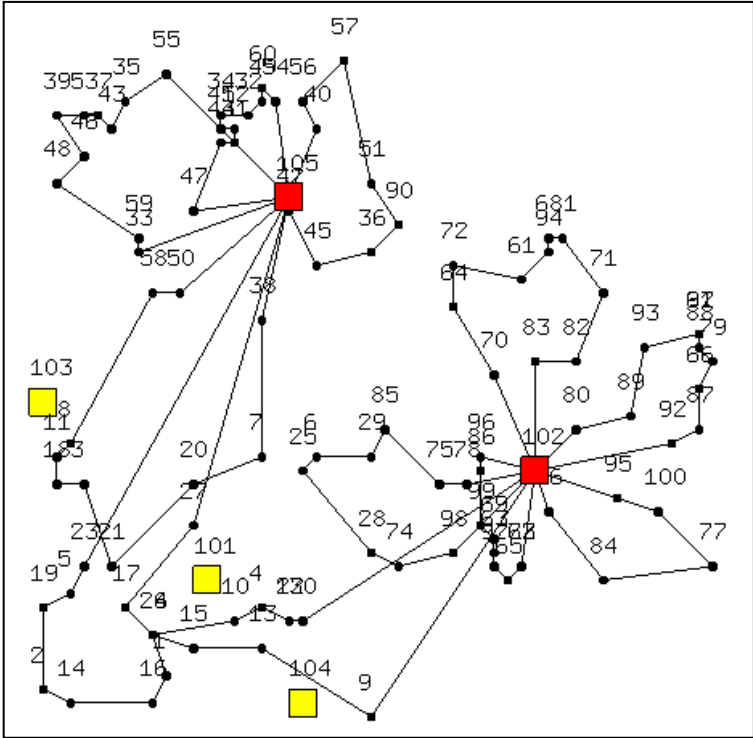


Custo Total: 193671,00 (Solução Ótima)

Rotas						
Rota 1	103	74	56	100	96	103
Rota 2	103	76	67	55	78	103
Rota 3	103	71	61	51	59	103
Rota 4	103	90	94	86	98	63103
Rota 5	103	85	64	57	52	103
Rota 6	105	50	75	58	12	6105
Rota 7	103	72	95	88	93	103
Rota 8	103	77	62	73	65	53103
Rota 9	103	89	84	60	79	99103
Rota 10	105	18	28	44	37	105
Rota 11	105	38	1	43	15	105
Rota 12	105	47	39	34	7	31105
Rota 13	105	10	24	40	36	105
Rota 14	105	42	3	41	11	105
Rota 15	105	17	45	48	25	13105
Rota 16	103	69	97	80	81	103
Rota 17	105	23	20	4	16	33105
Rota 18	103	91	83	68	54	103
Rota 19	103	87	92	70	66	82103
Rota 20	105	14	8	21	32	105
Rota 21	105	29	22	5	26	105
Rota 22	105	30	46	49	2	105
Rota 23	105	35	19	9	27	105

Capacidade Utilizada

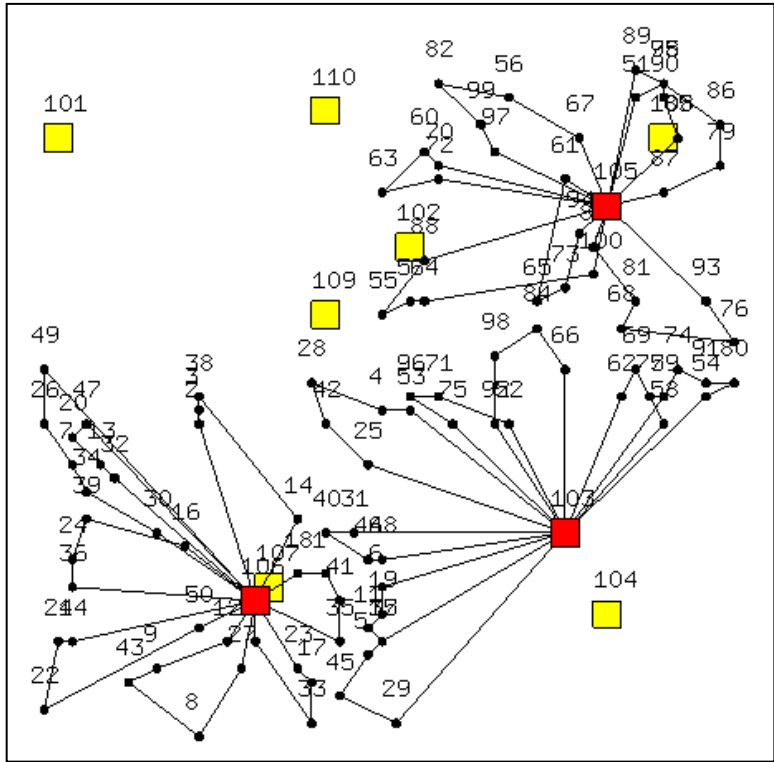
Rota 1	69	Depósitos 101	0
Rota 2	69	Depósitos 102	0
Rota 3	68	Depósitos 103	756
Rota 4	69	Depósitos 104	0
Rota 5	68	Depósitos 105	802
Rota 6	69		
Rota 7	69		
Rota 8	67		
Rota 9	70		
Rota 10	61		
Rota 11	59		
Rota 12	70		
Rota 13	70		
Rota 14	68		
Rota 15	70		
Rota 16	70		
Rota 17	70		
Rota 18	67		
Rota 19	70		
Rota 20	63		
Rota 21	68		
Rota 22	67		
Rota 23	67		



Custo Total: 152441,00 (Solução Ótima)

Rotas												
Rota 1	105	55	35	43	37	53	39	46	48	59	33	105
Rota 2	105	27	17	24	1	16	14	2	19	5	23	105
Rota 3	102	78	75	85	29	6	25	28	74	98	102	
Rota 4	102	80	89	93	91	67	88	79	66	87	92	102
Rota 5	102	96	86	99	69	63	97	65	73	62	102	
Rota 6	105	47	44	31	52	41	34	32	49	60	54	105
Rota 7	102	83	82	71	81	68	94	61	72	64	70	102
Rota 8	105	38	7	20	21	3	18	11	8	58	50	105
Rota 9	105	42	45	36	90	51	57	56	40	105		
Rota 10	102	30	22	12	4	10	26	15	13	9	102	
Rota 11	102	95	100	77	84	76	102					

Capacidade Utilizada			
Rota 1	148	Depósitos 101	0
Rota 2	150	Depósitos 102	825
Rota 3	150	Depósitos 103	0
Rota 4	147	Depósitos 104	0
Rota 5	146	Depósitos 105	737
Rota 6	149	Depósitos 106	0
Rota 7	148	Depósitos 107	
Rota 8	147	Depósitos 108	
Rota 9	143	Depósitos 109	
Rota 10	148	Depósitos 110	
Rota 11	86		

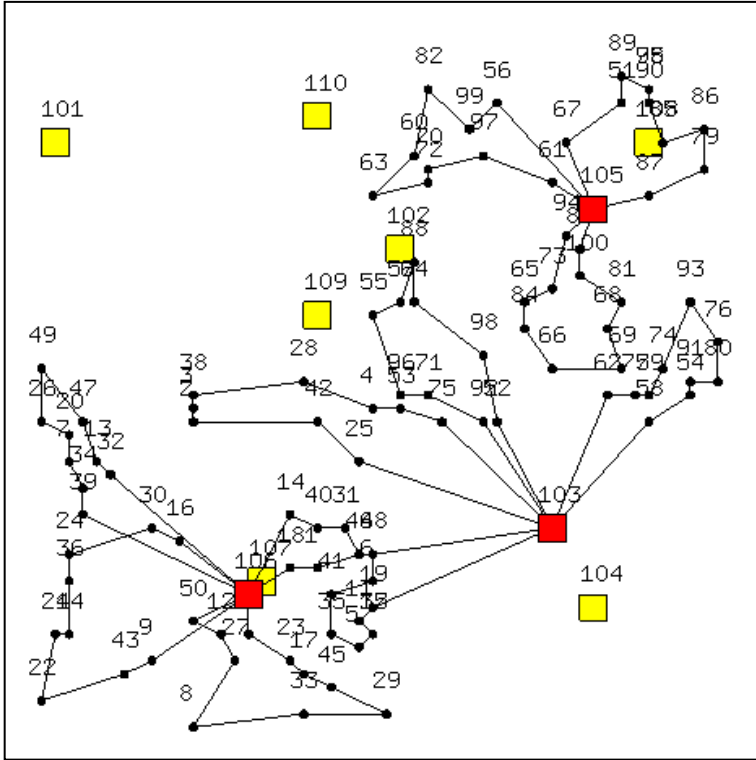


Custo Total: 243590,00 (Solução Ótima)

Rotas							
Rota 1	103	52	71	96	75	103	
Rota 2	105	94	73	65	61	105	
Rota 3	105	67	56	82	99	97	105
Rota 4	105	89	78	90	85	105	
Rota 5	103	59	74	91	80	54	103
Rota 6	106	50	22	21	44	106	
Rota 7	106	12	9	43	8	27	106
Rota 8	105	93	76	68	81	83	105
Rota 9	105	72	63	60	70	105	
Rota 10	103	62	69	77	58	103	
Rota 11	103	29	45	5	15	103	
Rota 12	103	53	4	28	42	25	103
Rota 13	103	6	19	11	37	103	
Rota 14	106	2	3	38	14	106	
Rota 15	106	32	13	20	47	106	
Rota 16	106	10	33	17	23	106	
Rota 17	103	66	84	98	92	103	
Rota 18	105	88	55	57	64	100	105
Rota 19	106	35	41	1	18	106	
Rota 20	106	36	24	39	16	106	
Rota 21	106	30	34	7	26	49	106
Rota 22	103	31	40	46	48	103	
Rota 23	105	51	95	86	79	87	105

Capacidade Utilizada

Rota 1	69	Depósitos 101	0
Rota 2	69	Depósitos 102	0
Rota 3	66	Depósitos 103	524
Rota 4	70	Depósitos 104	0
Rota 5	66	Depósitos 105	477
Rota 6	66	Depósitos 106	535
Rota 7	70	Depósitos 107	0
Rota 8	70	Depósitos 108	0
Rota 9	67	Depósitos 109	0
Rota 10	64	Depósitos 110	0
Rota 11	64		
Rota 12	70		
Rota 13	67		
Rota 14	68		
Rota 15	70		
Rota 16	61		
Rota 17	63		
Rota 18	68		
Rota 19	62		
Rota 20	68		
Rota 21	70		
Rota 22	61		
Rota 23	67		



Coord 100-10-2b

Custo Total: 203988,00 (Solução Ótima)

Rotas

Rota 1	103	75	53	4	28	38	3	2	42	25	103	
Rota 2	106	9	43	22	21	44	36	24	30	16	106	
Rota 3	105	67	51	89	78	95	90	85	86	79	87	105
Rota 4	106	39	34	7	20	26	49	47	13	32	106	
Rota 5	105	61	97	70	72	63	60	82	99	56	105	
Rota 6	106	10	23	17	45	29	33	8	27	12	50	106
Rota 7	103	92	71	96	55	57	88	64	98	52	103	
Rota 8	103	19	11	15	37	5	35	41	6	48	103	
Rota 9	103	58	54	91	80	76	93	74	59	77	62	103
Rota 10	105	94	73	65	84	66	69	68	81	100	83	105
Rota 11	106	18	1	46	31	40	14	106				

Capacidade Utilizada

Rota 1	137	Depósitos 101	0
Rota 2	150	Depósitos 102	0
Rota 3	150	Depósitos 103	560
Rota 4	144	Depósitos 104	0
Rota 5	139	Depósitos 105	438
Rota 6	145	Depósitos 106	538
Rota 7	139	Depósitos 107	0
Rota 8	140	Depósitos 108	0
Rota 9	144	Depósitos 109	0
Rota 10	149	Depósitos 110	0
Rota 11	99		