

TESE DE DOUTORADO Nº 256

**HEURÍSTICAS PARALELAS APLICADAS A PROBLEMAS DE  
ALOCÇÃO DE CONCENTRADORES**

**Rodrigo de Carvalho**

DATA DA DEFESA: 05/07/2017

Rodrigo de Carvalho

# **Heurísticas paralelas aplicadas a problemas de alocação de concentradores**

Tese de Doutorado submetida à Banca Examinadora designada pelo Colegiado do Programa de Pós Graduação em Engenharia Elétrica da Escola de Engenharia da Universidade Federal de Minas Gerais, como requisito para o título de Doutor em Engenharia Elétrica

Universidade Federal de Minas Gerais – UFMG  
Programa de Pós Graduação em Engenharia Elétrica  
Programa de Pós-Graduação

Orientador: Rodney Rezende Saldanha  
Coorientadores: Ricardo Saraiva de Camargo e Alexandre Xavier Martins

Brasil  
Julho de 2017

C331h

Carvalho, Rodrigo de.

Heurísticas paralelas aplicadas a problemas de alocação de concentradores [manuscrito] / Rodrigo de Carvalho. - 2017.  
165 f., enc.: il.

Orientador: Rodney Rezende Saldanha.

Tese (doutorado) Universidade Federal de Minas Gerais,  
Escola de Engenharia.

Bibliografia: f. 151-165.

1. Engenharia elétrica - Teses. 2. Otimização combinatória - Teses.  
3. Algoritmos paralelos - Teses. 4. Heurística - Teses. I. Saldanha,  
Rodney Rezende. II. Universidade Federal de Minas Gerais. Escola de  
Engenharia. III. Título.

CDU: 621.3(043)

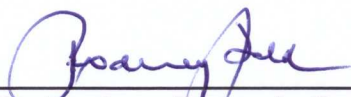
**"Heurísticas Paralelas Aplicadas a  
Problemas de Alocação de Concentradores"**

**Rodrigo de Carvalho**

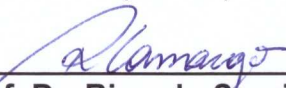
Tese de Doutorado submetida à Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em Engenharia Elétrica da Escola de Engenharia da Universidade Federal de Minas Gerais, como requisito para obtenção do grau de Doutor em Engenharia Elétrica.

Aprovada em 05 de julho de 2017.

Por:



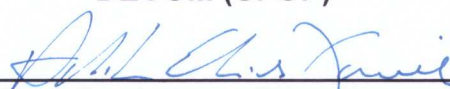
Prof. Dr. Rodney Rezende Saldanha  
DEE (UFMG) - Orientador



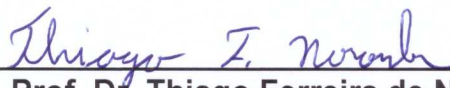
Prof. Dr. Ricardo Saraiva de Camargo  
DEP EE (UFMG) - Coorientador



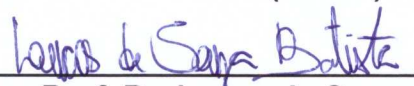
Prof. Dr. Marcone Jamilson Freitas Souza  
DECOM (UFOP)



Prof. Dr. Adilson Elias Xavier  
PESC/COPPE (UFRJ)



Prof. Dr. Thiago Ferreira de Noronha  
DCC (UFMG)



Prof. Dr. Lucas de Souza Batista  
DEE (UFMG)



*Este trabalho é dedicado às crianças adultas que,  
quando pequenas, sonharam em se tornar cientistas.*



# Agradecimentos

Gostaria primeiro de agradecer a minha mãe (in memoriam), Maria, sem o seu apoio e encorajamento eu não teria conseguido alcançar meus objetivos. “A dor de tê-la perdido não fez esquecer a alegria de tê-la tido e a esperança de reencontrá-la”.

Não menos importante, minha eterna gratidão ao meu pai Antônio e irmã Renata, porto seguro em todos os momentos. Não poderia esquecer da minha prima Flávia, que esteve presente nos momentos mais difíceis, muito obrigado!

Aos amigos do laboratório LOPAC, principalmente a Luciana "Mãe Temporária", André (Google) e Bruno, pelos momentos de descontração e suporte técnico no decorrer deste e de outros trabalhos.

Aos meus orientadores Dr. Rodney Rezende Saldanha, Dr. Ricardo Saraiva de Camargo e Dr. Alexandre Xavier Martins, pelas orientações, direcionamento deste trabalho e, principalmente, pela amizade.

Por fim, gostaria de agradecer à Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) pela concessão de bolsa de estudo por meio do Programa de Fomento à Pós-Graduação (PROF).



*In the scientific context: "This is not the end,  
it is not even the beginning of the end.  
But it is, perhaps, the end of the beginning."*



# Resumo

Projeto de redes do tipo Eixo-Raio (E-R), também conhecido como problemas de localização de concentradores, tem sua importância reconhecida em função da sua aplicação nas mais diversas áreas, tais como sistemas de transporte, telecomunicações e redes de serviço. Geralmente, estes problemas consistem em selecionar, entre um conjunto de possibilidades, nós concentradores que atuam como centros de comutação ou concentração de fluxo, permitindo assim a distribuição de mercadorias, informação e serviços entre pontos de uma rede. Este tipo de abordagem tem suas vantagens, pois dependendo do contexto da aplicação pode reduzir custos de infraestrutura, racionalizar o manuseio e triagem de fluxo, além de possibilitar a economia de escala por meio da consolidação de fluxos com o uso de tecnologias para o transporte ou transmissão de grandes volumes.

Este trabalho apresenta contribuições para alguns dos problemas de localização de concentradores ao propor e avaliar o desempenho de dois algoritmos heurísticos paralelos, utilizando apenas a CPU, para tratar diferentes variantes e cenários do problema de planejamento de redes do tipo E-R: problema de localização de concentradores não capacitados com alocação simples (USAHLP, do inglês *Uncapacitated single allocation hub location problem*); problema de localização de concentradores não capacitados com alocação múltipla (UMAHLP, do inglês *Uncapacitated multiple allocation hub location problem*); problema de localização de concentradores em anel não capacitados como alocação simples (CHLP, do inglês *Cycle Hub Location Problem*). Embora algoritmos heurísticos paralelos tenham sido aplicados a outros problemas de localização de concentradores, não foi encontrada nenhuma abordagem para os problemas tratados nesta tese.

Os algoritmos propostos neste trabalho foram comparados com os melhores algoritmos da literatura aplicados aos problemas estudados. Para isso, um conjunto de experimentos foi feito utilizando instâncias teste, com tamanho variando de 10 a 200 nós, e instâncias criadas para este trabalho, que chegam a ter 3000 nós. O estudo destes problemas com instâncias deste tamanho é importante devido a diversas aplicações: planejamento de redes de transporte aéreo, se considerado todos os agentes envolvidos, tais como cidades, aeroportos regionais, estaduais e nacionais; problemas de logística, considerando pontos de distribuição com diferentes propósitos em escala nacional.

Como esperado, os resultados obtidos mostram que as versões paralelas das heurísticas propostas superam, em tempo e qualidade, as versões sequenciais das mesmas. Além disso, foram obtidos resultados melhores que aqueles alcançados pelos algoritmos da literatura, mostrando a eficiência da proposta.



# Abstract

The design of hub-and-spoke networks represented by hub location problems is of great importance due to the presence of such topology in a myriad of applications such as logistics, telecommunication systems and service networks. Generally speaking, hub location problems consist of locating hubs nodes, allocating non-hub nodes to the installed hubs, and inter-connecting the hub network. The bundle of flows at the hubs allows the use of more efficient, high volume carriers between hubs achieving thus scale economies. Furthermore, depending on the application, these networks can reduce infrastructure costs.

The main contribution of this thesis is the proposal and evaluation of two parallel heuristics devised to solve three hub location problems: the uncapacitated single allocation hub location problem (USAHLP); the uncapacitated multiple allocation hub location problem (UMAHLP); and the cycle hub location problem. Although parallel heuristic algorithms have been applied to other hub location problems and as further as the author knows it is the first time such approach is sought for the present problems.

The proposed heuristics' performance is compared with well-known heuristics from the literature on solving the AP benchmark test instances, and a larger scale set specially crafted to resemble a Brazilian logistic context. The AP test instance sizes range from 10 to 200 nodes, whereas the proposed ones reach up to 3000 nodes.

As expected, the attained results show that the parallel heuristics outperform the sequential ones, both in time and solution quality for all tested problem. Finally, practical problems usually have a large amount of data which results in a high-dimensional search space. Thus the study of methods capable of dealing with such large scale problems are critical and a matter of importance to decision makers.



# Lista de ilustrações

Figura 1 – Exemplos de topologias de redes. Fonte: Adaptado de Camargo (2007)	20
Figura 2 – Média de velocidade dos processadores a partir do ano 2000. . . . .	24
Figura 3 – Exemplos de ótimo local e ótimo global . . . . .	34
Figura 4 – Exemplo de modelo baseado em Algoritmo não cooperativo . . . . .	37
Figura 5 – Topologias clássicas para troca de informações entre tarefas/processos.	38
Figura 6 – Combinação de três modelos de meta-heurísticas paralelas. Fonte: Adaptado de (TALBI, 2009) . . . . .	39
Figura 7 – Exemplo de escalonamento de processo em uma região crítica. Fonte: Adaptado de Tanenbaum (2008) . . . . .	41
Figura 8 – Meta-heurística ILS-RVND-PR . . . . .	43
Figura 9 – Posições disponíveis para aplicar o método de perturbação por <i>thread</i> .	45
Figura 10 – <i>thread 1</i> . . . . .	46
Figura 11 – <i>thread 2</i> . . . . .	46
Figura 12 – <i>thread 3</i> . . . . .	46
Figura 13 – Exemplo de um gráfico TTT . . . . .	48
Figura 14 – Ilustração das instâncias BR propostas neste trabalho. . . . .	51
Figura 15 – Exemplo de solução para o problema de localização de facilidades . . .	55
Figura 16 – Rede ponto-a-ponto x Rede baseado em concentradores . . . . .	57
Figura 17 – Rede E-R com a) alocação simples e b) alocação múltipla . . . . .	58
Figura 18 – Rede E-R Estrela-Estrela . . . . .	59
Figura 19 – Rede E-R Anel-Estrela . . . . .	60
Figura 20 – Rede E-R Anel-Anel . . . . .	60
Figura 21 – Ilustração de uma rede <i>backbone</i> /tributaria. . . . .	69
Figura 22 – Exemplo de uma rede do tipo E-R com alocação simples . . . . .	74
Figura 23 – Representação computacional para o problema de localização de concentradores com alocação simples. . . . .	80
Figura 24 – Exemplo de uma iteração do método de construção . . . . .	82
Figura 25 – Exemplo de uma iteração do método de construção . . . . .	84
Figura 26 – Ilustração do método <i>Path-Relinking</i> . . . . .	87
Figura 27 – <i>speedup</i> calculado para o método de construção paralelo em relação ao sequencial . . . . .	88
Figura 28 – Quantidade de avaliações de função e iterações para obter a solução inicial. . . . .	91
Figura 29 – Gráficos TTT. . . . .	102
Figura 30 – Gráficos TTT. . . . .	103

Figura 31 – Gráfico contendo informação de média e desvio padrão em relação ao Rank, obtido a partir do teste de Friedman e do procedimento de comparações múltiplas. Algoritmos: 1 - ILS-RVND2; 2 - ILS-RVND1; 3 - Busca Tabu. . . . .	104
Figura 32 – Exemplo de uma rede do tipo E-R com alocação múltipla . . . . .	108
Figura 33 – Representação computacional para o problema de localização de concentradores com alocação múltipla. . . . .	113
Figura 34 – <i>speedup</i> calculado para o método de construção paralelo em relação ao sequencial . . . . .	118
Figura 35 – Proporção de tempo gasto por $C1$ quando comparado a $C2$ para obter uma solução em relação ao número de nós . . . . .	119
Figura 36 – Gap em relação a função objetivo entre $C2$ e $C1$ . . . . .	119
Figura 37 – Gráficos TTT. . . . .	124
Figura 38 – Gráfico contendo informação de média e intervalo de confiança em relação ao Rank, obtido a partir do teste de Friedman e do procedimento de comparações múltiplas. . . . .	131
Figura 39 – Gráficos TTT. . . . .	131
Figura 40 – Gráficos TTT. . . . .	133
Figura 41 – Exemplo de uma rede do tipo E-R com alocação simples e concentradores ligados em anel . . . . .	137
Figura 42 – Representação computacional para o problema de localização de concentradores em anel com alocação simples. . . . .	139
Figura 43 – Ilustração do método <i>Path-Relinking</i> aplicado ao problema de localização de concentradores em anel com alocação simples. . . . .	142
Figura 44 – Função objetivo obtida e tempo demandado para construir uma solução utilizando diferentes métodos. Construção proposta (Preto) × Construção da Literatura (Branco). . . . .	143
Figura 45 – Função objetivo obtida pelos diferentes métodos utilizados neste trabalho. Métodos: GRASP (Preto), ILS-RVND1 (Cinza) e ILS-RVND2 (Branco). . . . .	144
Figura 46 – Função objetivo obtida pelos diferentes métodos utilizados neste trabalho para $n \leq 100$ . Métodos: GRASP (Preto), ILS-RVND1 (Cinza) e ILS-RVND2 (Branco). . . . .	144
Figura 47 – Gráfico contendo informações de média e intervalo de confiança em relação ao Rank, obtido a partir do teste de Friedman e do procedimento de comparações múltiplas. Algoritmo: 1 - GRASP; 2 - ILS-RVND1; 3 - ILS-RVND2 . . . . .	145
Figura 48 – Gráficos TTT. . . . .	146
Figura 49 – Gráficos TTT. . . . .	147

# Lista de tabelas

Tabela 1 – Número de avaliações de função para se obter uma solução para alguns problemas tratados neste trabalho. . . . .	90
Tabela 2 – Tempo médio em segundos para criação de uma solução utilizando os métodos de construção paralelo e sequencial. . . . .	92
Tabela 3 – Valor de função objetivo obtido pelos métodos de construção 1 e 2 . . .	94
Tabela 4 – Speedup obtido com diferentes configurações do algoritmo ILS-RVND1 e <i>hardware</i> . . . . .	95
Tabela 5 – <i>speedup ILSRVND2</i> . . . . .	95
Tabela 6 – Comparação entre os métodos heurísticos . . . . .	97
Tabela 7 – Valor de função objetivo obtido pelas diferente heurísticas após 0,4n segundos . . . . .	100
Tabela 8 – Número de operações para obter o menor caminho entre todos os pontos de um grafo. . . . .	115
Tabela 9 – Valor da função objetivo obtida pelas variações dos métodos de construção propostos. . . . .	120
Tabela 10 – Valor da função objetivo obtida pelas variações dos métodos de construção propostos. . . . .	121
Tabela 11 – Tempo gasto em segundos para obter uma solução pelas variações dos métodos de construção propostos. . . . .	122
Tabela 12 – Valores obtidos para alguns indicadores referentes aos métodos ILS-RVND1 e ILS-RVND2 . . . . .	123
Tabela 13 – Valores médios de função objetivo obtidos pelas variações dos métodos propostos neste trabalho (ILS-RVND1 e ILSRVND2). . . . .	126
Tabela 14 – Valores obtidos para alguns indicadores para métodos da literatura e o algoritmo proposto neste trabalho . . . . .	127
Tabela 15 – Valores de função objetivo obtidos pelo métodos da literatura e o algoritmo proposto neste trabalho. . . . .	129
Tabela 16 – Valores de função objetivo obtidos pelo métodos da literatura e o algoritmo proposto neste trabalho. . . . .	130
Tabela 17 – Valores de DevMin, DevMed e KBS obtidos para os métodos GRASP, ILS-RVND1 e ILS-RVND2 . . . . .	145



# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>19</b>
<b>1.1</b>	<b>Objetivo do trabalho</b>	<b>25</b>
1.1.1	Objetivo Geral	25
1.1.2	Objetivo Específico	25
<b>1.2</b>	<b>Contribuições</b>	<b>25</b>
<b>1.3</b>	<b>Organização do trabalho</b>	<b>26</b>
<b>2</b>	<b>CONCEITOS PRELIMINARES</b>	<b>29</b>
<b>2.1</b>	<b>Otimização Combinatória</b>	<b>29</b>
2.1.1	Métodos Exatos	30
2.1.2	Heurísticas	32
<b>2.2</b>	<b>Metaheurísticas Paralelas</b>	<b>35</b>
2.2.1	Arquitetura do Computador	39
2.2.2	PThreads e OpenMP	41
<b>2.3</b>	<b>Proposta (ILS-RVND-PR)</b>	<b>42</b>
<b>2.4</b>	<b>Medindo o desempenho dos métodos heurísticos</b>	<b>47</b>
<b>2.5</b>	<b>Instâncias</b>	<b>50</b>
<b>2.6</b>	<b>Conclusão</b>	<b>52</b>
<b>3</b>	<b>REVISÃO BIBLIOGRÁFICA - REDES BASEADAS EM CONCEN- TRADORES</b>	<b>53</b>
<b>3.1</b>	<b>Problema de localização de Facilidades</b>	<b>53</b>
<b>3.2</b>	<b>Redes do tipo Eixo-Raio</b>	<b>56</b>
<b>3.3</b>	<b>Aplicações para redes E-R</b>	<b>64</b>
3.3.1	Transporte Aéreo	64
3.3.2	Transporte público urbano	66
3.3.3	Telecomunicação	68
3.3.4	Transporte de carga terrestre	70
<b>3.4</b>	<b>Conclusão</b>	<b>72</b>
<b>4</b>	<b>REDES DO TIPO EIXO-RAIO COM ALOCAÇÃO SIMPLES</b>	<b>73</b>
<b>4.1</b>	<b>Introdução</b>	<b>73</b>
<b>4.2</b>	<b>Metodologia</b>	<b>80</b>
4.2.1	Representação Computacional	80
4.2.2	Método de construção	81
4.2.2.1	Método 1 ( <i>M1</i> )	81

4.2.2.2	Método 2 ( <i>M2</i> ) . . . . .	83
4.2.3	Estruturas de vizinhança . . . . .	84
4.2.4	Perturbação . . . . .	86
4.2.5	Path-Relinking . . . . .	86
<b>4.3</b>	<b>Experimentos Computacionais . . . . .</b>	<b>87</b>
4.3.1	Método de construção . . . . .	88
4.3.2	Influência do número de <i>threads</i> no comportamento do algoritmo ILS-RVND1	93
4.3.3	Influência do número de <i>threads</i> no comportamento do algoritmo ILS-RVND2	95
4.3.4	Análise dos componentes do ILS-RVND1 e ILS-RVND2 . . . . .	96
4.3.5	Proposta × Literatura . . . . .	97
<b>4.4</b>	<b>Conclusão . . . . .</b>	<b>105</b>
<b>5</b>	<b>REDES DO TIPO EIXO-RAIO COM ALOCAÇÃO MÚLTIPLA . . . . .</b>	<b>107</b>
<b>5.1</b>	<b>Introdução . . . . .</b>	<b>107</b>
<b>5.2</b>	<b>Metodologia . . . . .</b>	<b>112</b>
5.2.1	Representação Computacional . . . . .	112
5.2.2	Avaliação de Solução . . . . .	113
5.2.3	Método de construção . . . . .	114
5.2.4	Estruturas de vizinhança . . . . .	116
<b>5.3</b>	<b>Perturbação e <i>Path-Relinking</i> . . . . .</b>	<b>117</b>
<b>5.4</b>	<b>Experimentos Computacionais . . . . .</b>	<b>117</b>
5.4.1	Métodos de Construção . . . . .	117
<b>5.5</b>	<b>Comparação entre ILS-RVND1 × ILS-RVND2 . . . . .</b>	<b>123</b>
<b>5.6</b>	<b>Comparação com a Literatura . . . . .</b>	<b>127</b>
<b>5.7</b>	<b>Conclusão . . . . .</b>	<b>131</b>
<b>6</b>	<b>REDES DO TIPO EIXO-RAIO EM ANEL COM ALOCAÇÃO SIM- PLES . . . . .</b>	<b>135</b>
<b>6.1</b>	<b>Metodologia . . . . .</b>	<b>138</b>
<b>6.2</b>	<b>Representação Computacional . . . . .</b>	<b>138</b>
6.2.1	Avaliação da Solução . . . . .	139
6.2.2	Método de Construção . . . . .	140
6.2.3	Estruturas de Vizinhança . . . . .	140
6.2.4	Perturbação . . . . .	141
6.2.5	<i>Path-Relinking</i> . . . . .	141
<b>6.3</b>	<b>Experimentos Computacionais . . . . .</b>	<b>142</b>
<b>6.4</b>	<b>Conclusão . . . . .</b>	<b>147</b>
<b>7</b>	<b>CONCLUSÕES FINAIS E TRABALHOS FUTUROS . . . . .</b>	<b>149</b>
	<b>REFERÊNCIAS . . . . .</b>	<b>151</b>

# 1 Introdução

O transporte de carga e passageiros se configura como um serviço fundamental para o desenvolvimento de um país. Investimentos nesta área proporcionam ganhos nos mais diferentes setores, tais como econômico, social e ambiental. Um dos pontos positivos ao se aumentar da mobilidade é permitir desenvolver o comércio e, conseqüentemente, as atividades produtivas, além de diminuir as assimetrias regionais e melhorar a qualidade de vida e o bem-estar da população. Porém, projetar, expandir ou melhorar este serviço não é uma tarefa trivial e, seja na esfera pública ou privada, merece a devida importância. Neste contexto, criar metodologias e ferramentas que sejam capazes de auxiliar gestores na criação de redes de transportes eficientes é um grande desafio, considerando qualidade, robustez e tempo para se obter uma solução.

Atualmente, um dos recursos que tem sido empregado para projetar redes de transportes eficientes é a sua modelagem a partir de redes de concentradores. Este tipo de rede também é conhecido como redes do tipo Eixo-Raio (E-R) e, ao invés de conectar pontos de demanda diretamente, substitui essas conexões por uma rede de concentradores. Esses concentradores permitem que o tráfego seja agrupado e conduzido por meio de um sistema de transporte compartilhado, para então ser entregue aos seus destinos. Assim, as demandas entre muitos pares de origem-destino usufruem da economia de escala obtida ao se concentrar o fluxo durante o transporte e/ou transmissão. Essa economia de escala permite uma significativa redução nos custos de transporte e, conseqüentemente, faz com que os recursos sejam melhor utilizados.

A Figura 1 ilustra dois tipos diferentes de topologias de rede, onde (a) mostra uma rede ponto a ponto, na qual todos os pares de origem-destino são conectados. Já a parte (b) da Figura 1 ilustra uma rede do tipo E-R tradicional. É perceptível que o número de conexões é significativamente inferior na rede E-R, o que proporciona uma redução de custos operacionais. Dependendo do contexto da aplicação da rede há também o benefício da redução dos custos com infraestrutura.

A Figura 1 (b) também ilustra os principais componentes de uma rede do tipo E-R. Os triângulos representam os concentradores, os pontos representam os pontos de origem e destino (um mesmo ponto possui estes dois papéis), as linhas mais grossas entre os concentradores representam as conexões que possuem economia de escala. Esta economia é obtida devido o compartilhamento do meio de transporte, isto é, o custo por unidade transportada torna-se menor ao se aumentar o volume do tráfego através do mesmo meio de transporte.

A ideia do uso compartilhado de transporte, além de ser positivo em relação

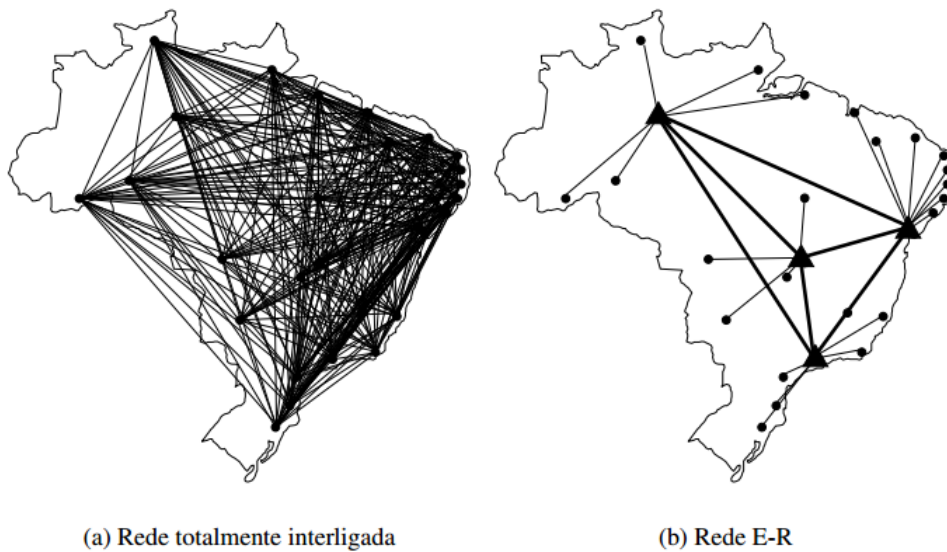


Figura 1 – Exemplos de topologias de redes. Fonte: Adaptado de [Camargo \(2007\)](#)

a diminuição de custos, pode contribuir para um menor impacto no meio ambiente. Isso devido ao fato que menos veículos em circulação pode proporcionar uma menor quantidade de gases sendo emitidos. Para se ter uma ideia do impacto da má utilização, e/ou planejamento no setor de transporte público, o crescimento de casos de problemas de saúde decorrentes da poluição gerada pelo tráfego, nas diferentes escalas, tem aumentado exponencialmente nos países em desenvolvimento. Se tratando de Brasil, cerca de 50% dos gases que causam o efeito estufa são emitidos pelo setor de transporte. Assim sendo, a mobilidade baseada no uso de meios de deslocamento coletivos é essencial para reduzir este número. Por sua vez, esse sistema tem que ser eficiente para que possa atrair cada dia mais usuários. Neste caso, pode se entender como eficiente uma rede de transporte que cubra o maior número de regiões, que tenha disponibilidade (número de usuários e transportadores equivalentes), que ofereça segurança e seja eficiente em relação a tempo de deslocamento.

Atender a todas as questões observadas anteriormente dá origem a um problema complexo que requer uma considerável quantia de recursos financeiros e um esforço significativo para gerenciá-lo. Redes do tipo concentradores podem ser usadas para auxiliar na resolução desse tipo de problema. Nesse contexto, os concentradores são conhecidos como estações de embarque e desembarque (mêtro ou BRT), enquanto os não concentradores podem ser a demanda de uma determinada região. A ideia principal é criar corredores de descolamento entre concentradores (terminais) que utilizem meios de transportes com maior capacidade e, a partir desses, atender os pontos de demanda usando linhas locais com capacidade inferior ([SÁ, 2015](#)). Vale ressaltar que, no planejamento deste tipo de rede se por um lado o objetivo é oferecer uma boa qualidade de serviço, por outro, as empresas responsáveis por fornecer tal estrutura querem gastar o mínimo possível, sendo importante

o uso dos recursos disponíveis de forma eficiente.

Já no setor de transporte de carga, a oferta inadequada de infraestrutura provoca entraves e gastos desnecessários gerando perdas de competitividade dos serviços ou produtos. Assim, um bom planejamento da logística torna-se indispensável para que seja possível minimizar os custos operacionais. Nos dias atuais as empresas deixaram de ver o transporte como uma simples atividade de movimentação, tornando-o um elo de ligação estratégico entre clientes e fornecedores. Neste contexto, o uso de metodologias que fazem o uso de redes do tipo E-R pode contribuir efetivamente no planejamento destas redes.

Essas empresas convivem diariamente com a pressão existente em relação a diminuição de custos, por outro lado, precisam manter o alto nível de qualidade do serviço. Assim, o sucesso delas está diretamente ligado a estes aspectos, que por sua vez tem ligação direta com o planejamento das suas redes de transporte. Diversos autores modelaram o projeto de redes para esse tipo de empresa baseado em redes de concentradores (Taylor et al. (1999), Camargo (2007), Cunha e Silva (2007)), com diferentes restrições e perspectivas. Por exemplo, empresas que trabalham com transporte de carga fracionada geralmente necessitam dispor de estruturas de apoio, localizadas em pontos estratégicos, para consolidar cargas de diferentes origens que serão entregues a regiões específicas. Um dos benefícios da consolidação da carga é a diminuição de custos de transporte nas transferências de longa distância, uma vez que a transferência dessa carga com o uso de um veículo de grande capacidade proporciona isto. Neste contexto, as estruturas de apoio são os concentradores, enquanto estruturas de apoio menores, ou clientes, são pontos de demanda.

Outro setor muito importante para desenvolvimento e economia de um país é o de transporte aéreo. No caso do Brasil, o assunto é de grande relevância, pois considerando as deficiências nos ramos ferroviário e rodoviário, o modal aéreo se torna o único meio de acesso viável a determinadas localidades. Além disso, o setor possibilita o crescimento da economia, uma vez que permite o fluxo de agentes de negócios e o escoamento produtivo, bem como viabiliza o turismo nacional, entre outros benefícios. Para que isso aconteça, o planejamento da malha aérea deve ser feito de forma eficiente.

Redes do tipo concentrador são aplicadas neste setor desde o Ato de Desregulamentação do transporte aéreo de 1978 nos Estados Unidos, que permitiu as companhias aéreas escolherem suas rotas e tarifas que seriam praticadas. Estas mudanças tiveram efeitos profundos sobre muitos aspectos da operação aérea daquele país, particularmente no que se refere a tarifas, nível de serviço, qualidade e segurança. Mas talvez o mais importante foi que as empresas aéreas alteraram a estrutura de suas rotas, desenvolvendo redes do tipo Eixo-Raio (SILVA, 2004). O impacto da utilização desse tipo de rede sobre a rentabilidade das companhias aéreas desse país foi estudado por Toh e Higgins (1985). Diversos outros estudos, aplicando este tipo de rede no planejamento do transporte aéreo foram feitos em diferentes lugares do mundo após isso.

É possível observar que redes E-R têm se destacado não apenas por permitir a melhor utilização dos recursos, mas também pela diversidade de sua aplicação, tais como transporte logístico, tanto transporte de cargas quanto passageiros, e problemas de redes de telecomunicações. Segundo [Silva \(2004\)](#), o desenvolvimento deste tipo de rede é uma das inovações mais importantes na indústria, e reconhecida como a sétima melhor ideia na série "Grandes Ideias da Década em Marketing" pela Associação Americana de Marketing nos anos 80.

De forma geral, o planejamento das redes do tipo Eixo-Raio consiste basicamente em definir os nós concentradores, chamados de eixos, e alocar os pontos de origem e destino aos concentradores, formando os raios. Originalmente, o problema de projetar uma rede do tipo eixo-raio considera que existem conexões entre todos os concentradores, e cada nó não concentrador pode estar ligado a um ou mais concentradores. Porém, para projetar alguns tipos de redes esses pressupostos não são indicados. Um bom exemplo são as redes de transporte público urbano, onde restrições físicas ou o alto custo de instalação podem inviabilizar a criação de conexões entre todos os nós concentradores.

Para contornar tal problema, suposições mais flexíveis foram propostas com a finalidade de ampliar a aplicabilidade de redes do tipo eixo-raio a diferentes áreas. Por exemplo, desconsiderar que existem conexões diretas entre cada par de concentradores; permitir conexões diretas entre nós não concentradores; particionar o problema em sub-redes, dentre outros. Outras variantes para o problema de planejamento de redes do tipo eixo-raio podem ser vistos em [Campbell \(1994b\)](#), [Campbell \(1994a\)](#), [Campbell, Ernst e Krishnamoorthy \(2002\)](#) e [Alumur e Kara \(2008\)](#).

Neste trabalho serão abordadas três variantes para o problema de planejamento de redes do tipo eixo-raio não capacitado. A primeira consiste no problema com alocação simples, onde é considerado que existem conexões entre todos os pares de concentradores, e cada nó não concentrador está ligado a apenas um concentrador. Nessa variante os custos de instalação de um concentrador e os de transporte são considerados, além disso, o número de concentradores não é conhecido a priori. Este problema é conhecido na literatura como Problema de localização de concentradores não capacitado com alocação simples, em inglês como *Uncapacitated single allocation hub location problem* (USAHLP). A segunda variante difere da primeira, pois é permitido que um nó não concentrador esteja alocado a mais de um nó concentrador. Na literatura o problema é conhecido como Problema de localização de concentradores não capacitado com alocação múltipla, em inglês como *Uncapacitated multiple allocation hub location problem* (UMAHLP). Finalmente, a terceira variante, conhecida como Problema de localização de concentradores em anel com alocação simples, em Inglês como *Cycle Hub Location Problem* (CHLP), consiste em ligar os concentradores por meio de um ciclo, nesse caso não existe ligação direta entre todos os pares de concentradores. Além disso, os nós não concentradores devem ser alocados

---

a somente um concentrador e é necessário definir a rota entre origem e destino. Nessa variante os custos de instalação de concentradores não são considerados e o número de concentradores é conhecido a priori. Em ambos os casos o objetivo é um projetar redes do tipo E-R de forma otimizada, em outras palavras, encontrar uma solução onde com o menor custo possível.

Os problemas estudados neste trabalho são de grande dimensão e, além disso, estão inseridos na classe de problemas NP-Difícil (CORMEN et al., 1990). Para tais problemas, não é conhecido nenhum método determinístico (algoritmo computacional) capaz de encontrar a solução ótima com complexidade de tempo polinomial independente da dimensão da instância a ser resolvida. Além disso, problemas dessa natureza possuem uma condição inerente a certos tipos de problemas matemáticos nos quais um pequeno aumento no tamanho do problema (número de elementos de dados ou parâmetros operacionais) pode levar a um imenso aumento no tempo necessário para a obtenção da resposta. Desse modo, propor metodologias que resolvam problemas com estas características, de forma eficiente, ainda é um grande desafio.

Diferentes abordagens têm sido propostas na literatura para resolver problemas desta natureza, dentre as quais podemos destacar duas classes de métodos: Exatos e Heurísticos. O primeiro, embora garanta a otimalidade da solução, em problemas de grande escala quase não são aplicados devido ao alto custo computacional. Já as heurísticas, embora não tenham nenhuma garantia de qualidade da solução, têm se mostrado mais eficientes para resolver este tipo de problema por retornar soluções de qualidade em tempo razoável. Vale ressaltar que esforços têm sido empregados nestas duas frentes com o objetivo de aumentar a escalabilidade e eficiência desses métodos. Este trabalho se limita ao desenvolvimento de métodos heurísticos para os problemas abordados.

Há, no desenvolver das heurísticas utilizadas neste trabalho, a preocupação em utilizar recursos computacionais de forma eficiente, mais especificamente o poder de processamento das atuais estações de trabalho (PCs). Atualmente a frequência (Hz) dos novos processadores não está aumentando muito e a razão para isso é chamada eletromigração. Este fenômeno pode ocorrer quando processadores operam acima das frequências permitidas ou com uma tensão mais alta que a recomendada. Para chips semicondutores o problema surge quando o tamanho do circuito diminui e a frequência aumenta em um determinado ponto. O resultado da eletromigração é a redução do tempo de vida de um processador, ou no pior caso, sua inutilização. Para evitar esse fenômeno, os processadores passaram a contar com um número maior de núcleos, ao invés de ser aumentada sua frequência. Dessa forma, foi possível aumentar o poder de processamento dos computadores, diminuindo problemas com a eletromigração. A Figura 2 mostra que nos últimos anos os ganhos de velocidade de operação dos processadores são cada vez menores.

Assim sendo, para utilizar de forma eficiente os PCs atuais, é necessário que se utilize os vários núcleos disponíveis para dividir a carga de processamento e, então, obter ganho de "potência". A partir de uma revisão da literatura, foi possível perceber que métodos heurísticos que utilizam o recurso de processamento paralelo, tanto em tecnologias mais desenvolvidas, como *cluster* de processamentos, ou mesmo em computadores multi-*threads*, ainda são pouco exploradas. Para os problemas tratados neste trabalho é praticamente inexistente o uso deste recurso. Assim, são aqui propostas heurísticas que utilizam este recurso, utilizando apenas a CPU, para melhorar o desempenho em relação a tempo computacional e qualidade das soluções. Vale ressaltar que, embora seja de conhecimento de tecnologias promissoras para tal abordagem, tal como GPU, o objetivo aqui é mostrar que não é necessária a utilização de tecnologias proprietárias, ou mais complexas, para obter ganhos significativos.

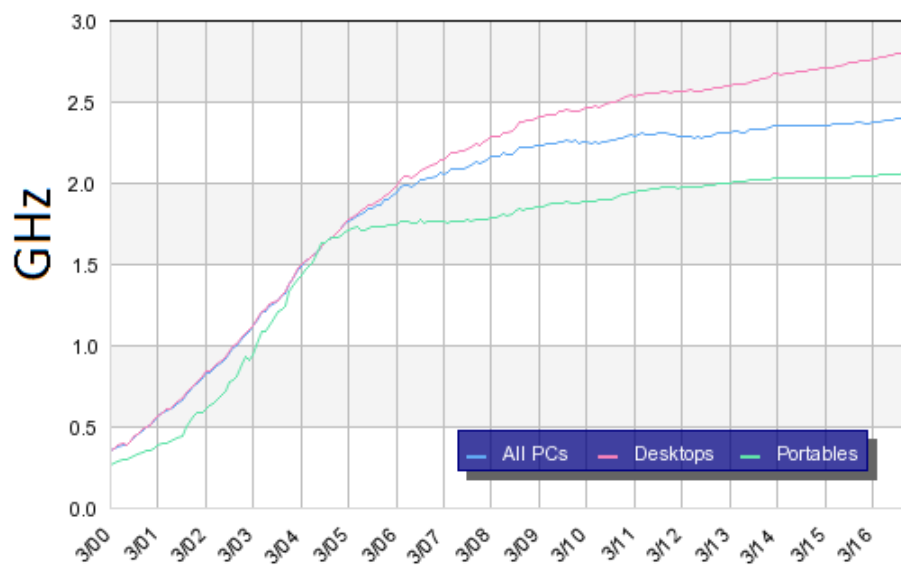


Figura 2 – Média de velocidade dos processadores a partir do ano 2000.

Ao desenvolver tais métodos, surgem perguntas como "Que parte do algoritmo deve ser executado em paralelo?", "A paralelização dará o resultado desejado?" ou "As tarefas devem comunicar-se umas com as outras ou não?". As respostas para estas perguntas podem não ser triviais, ainda mais se considerarmos que o mal uso deste recurso pode impactar negativamente no desempenho do algoritmo, tornando-o menos eficiente quando comparado com um algoritmo sequencial (BAEVRE; GJENGSTO, 2009). Assim sendo, para chegar nas propostas desta tese foram feitos vários experimentos com o objetivo de obter uma proposta simples e robusta.

## 1.1 Objetivo do trabalho

### 1.1.1 Objetivo Geral

A presente tese tem como objetivo principal explorar três variantes do projeto de redes Eixo-Raio, evidenciando suas aplicações em diferentes contextos e propondo heurísticas que exploram o paralelismo das atuais estações de trabalho para solucionar as diferentes versões abordadas de maneira satisfatória.

### 1.1.2 Objetivo Específico

Para atingir o objetivo geral do trabalho, os seguintes objetivos específicos são definidos.

- Explorar na literatura as diferentes abordagens para o projeto de redes do tipo eixo-raio.
- Propor heurísticas eficientes, utilizando o recurso de paralelização em CPU, para as diferentes abordagens do problema de projeto de redes do tipo eixo-raio.
- Validar os resultados produzidos pelos métodos com a utilização de ferramentas estatísticas;
- Propor instâncias maiores que as utilizadas na literatura, com o objetivo de testar a escalabilidade dos métodos. Normalmente, são utilizadas instâncias de até 400 nós para testar o desempenho de diferentes metodologias. Neste trabalho são propostas instâncias com até 3000 nós.

## 1.2 Contribuições

O projeto de redes do tipo E-R, nas suas diferentes variantes, é um problema que têm sido amplamente estudado nos últimos anos. Muitas pesquisas têm focado no desenvolvimento de algoritmos mais eficientes e eficazes para resolução do problema.

Este trabalho contribui com a proposta de uma metaheurística paralela, utilizando apenas a CPU, para resolver três variantes do projeto de rede E-R, sendo elas: Problema de localização de concentradores não capacitado com alocação simples; Problema de localização de concentradores não capacitado com alocação múltipla; Problema de localização de concentradores em anel com alocação simples. Estes problemas contemplam algumas características específicas que os aproximam de diferentes aplicações reais. Além disso, as instâncias utilizadas neste trabalho são significativamente maiores do que as encontradas na literatura.

Foram criadas duas variantes da metaheurística proposta, utilizando diferentes estratégias de paralelização. Observa-se que ambas podem ser facilmente adaptadas para resolver diferentes problemas e não possuem parâmetros a serem ajustados. Os resultados obtidos mostraram a eficiência das propostas em relação aos algoritmos da literatura.

### 1.3 Organização do trabalho

O restante deste trabalho está organizado da seguinte forma:

O Capítulo 2 apresenta alguns conceitos básicos sobre problemas de otimização combinatória e métodos que são aplicados a esta classe de problemas. Este Capítulo descreve também a meta-heurística que é utilizada para resolver as diferentes variantes dos problemas de planejamento de redes do tipo Eixo-Raio. Finalmente, são apresentadas as instâncias e indicadores utilizados no decorrer do trabalho.

No Capítulo 3 é feita uma breve revisão bibliográfica mostrando como os problemas de alocação de concentradores, bem como os métodos aplicados a estes problemas foram evoluindo no decorrer dos anos. O objetivo não é discutir todos os trabalhos, mas aqueles que tiveram maiores contribuições para a área e para este trabalho.

O Capítulo 4 aborda o projeto de redes do tipo Eixo-Raio tradicional, mais especificamente, aquele onde as conexões entre os concentradores são completas, existe custo fixo de instalação de concentradores e a alocação de nós não concentradores a concentradores é simples, ou seja, cada nó não concentrador pode estar alocado a apenas um concentrador. Além do problema, este Capítulo contempla os detalhes da heurística utilizada, bem como os experimentos computacionais e conclusões obtidas. Nos experimentos computacionais é verificado o ganho de desempenho ao paralelizar a heurística proposta. Além disso, os parâmetros obtidos neste Capítulo serão utilizados para resolver diferentes variantes do problema de alocação de concentradores no restante do trabalho. O objetivo é verificar o comportamento do algoritmo sem maiores personalizações.

No Capítulo 5 é tratado o problema de alocação de concentradores com alocação múltipla e custos fixos. Este problema é similar ao problema abordado no Capítulo 4, porém é possível que um nó não concentrador esteja alocado a mais de um concentrador. Novamente, a heurística, experimentos computacionais e resultados são contemplados.

O Capítulo 6 aborda o problema conhecido como *Cycle Hub Location Problem*, no qual os nós concentradores são organizados em forma de anel e, diferentemente dos problemas abordados anteriormente, não existe ligação direta entre todos os concentradores. Além disso, não é computado o custo de instalação e as alocações dos nós aos concentradores são simples. Além da formulação do problema, este Capítulo contempla a descrição da heurística utilizada, bem como os experimentos computacionais e conclusões obtidas.

O Capítulo 7 apresenta as principais conclusões e trabalhos futuros que podem ser realizados como sequência deste trabalho.



## 2 Conceitos Preliminares

Neste capítulo serão apresentados alguns conceitos e definições necessárias para entender o restante deste trabalho. Inicialmente será descrito o que é um problema de otimização combinatória. Em seguida, são apresentadas as metodologias mais comuns para solução desta classe de problemas. Também será apresentado uma breve revisão bibliográfica sobre algoritmos de otimização, mas especificamente das estratégias de paralelização utilizadas em meta-heurísticas, bem como uma discussão sobre as principais limitações e vantagens deste tipo de abordagem. A meta-heurística, indicadores utilizados para verificar o desempenho da mesma e as instâncias utilizadas no decorrer deste trabalho também serão detalhados.

### 2.1 Otimização Combinatória

O termo "otimização" pode ser entendido como o processo através do qual se obtém o melhor valor de uma grandeza. Já a análise combinatória, consiste no estudo matemático de arranjos analisando possibilidades e combinações. Tradicionalmente, os combinatorialistas têm se preocupado com questões de existência ou de enumeração, algumas das principais perguntas para estes estudiosos são "existe um tipo particular de arranjo?" ou "quantos desses arranjos existem?". A junção destes dois termos deu nome a uma das classes mais importantes de problemas computacionais, os chamados problemas de otimização combinatória. Isso porque ao invés de analisar possibilidades ou combinações específicas, os estudiosos passaram a procurar, entre um conjunto finito de possibilidades, qual o melhor arranjo que atendia determinadas regras, também chamadas de restrições.

A otimização, de forma geral, é o ramo da matemática que engloba o estudo quantitativo do ótimo e os métodos usados para encontrar o ótimo. Esta teoria posta em prática é definida pela coleção de técnicas, métodos, procedimentos e algoritmos que podem ser usados para a localização do ótimo de uma função. Quando se trata de otimização combinatória, pode se dizer que ela está envolvida com modelos e métodos de otimização sobre escolhas discretas. Sua base teórica está amparada na teoria da programação linear, e tem ligações fortes com as seguintes áreas: matemática discreta; teoria da probabilidade; ciência da computação, mais especificamente em algoritmos e complexidade.

Formalmente um problema desta classe pode ser descrito como: Dado um conjunto finito e discreto  $\mathbb{E} = \{e_1, e_2, \dots, e_n\}$ , um problema de otimização combinatória é definido sobre todos os subconjuntos viáveis de  $\mathbb{E}$ , denotado por  $\mathcal{F}$ , em que  $\mathcal{F}$  também é discreto e finito. Ainda, considerando uma função objetivo  $f(.) : \mathcal{F} \rightarrow \mathbb{R}$  e uma solução  $\mathbb{S} \subseteq \mathbb{E}$ , um problema de otimização combinatória pode ser formulado de acordo como (2.1), onde  $\mathbb{S}^*$  é

chamada de solução ótima (WOLSEY, 1998).

$$\text{Encontrar } \mathbb{S}^* = \arg \min_{\mathbb{S} \subseteq \mathbb{E}} \{f(\mathbb{S}) : \mathbb{S} \in \mathcal{F}\} \quad (2.1)$$

A princípio, uma solução ótima poderia ser obtida por meio de uma simples enumeração, mas, na prática, frequentemente isto se torna inviável, devido ao número extremamente alto de soluções possíveis. Nesse caso, mesmo que se utilize um supercomputador, o tempo de processamento poderia ser de várias semanas ou até anos.

Com o objetivo de ilustrar a situação descrita anteriormente, considere que o número de soluções possíveis para um problema de agrupamento com  $a$  elementos em  $b$  grupos seja dado pela equação (2.2). Se  $a$  e  $b$  assumem os respectivos valores 100 e 10, o número de soluções possíveis seria  $2.75 \times 10^{93}$ . Avaliar essa quantidade de soluções, em busca da melhor, demandaria muito tempo computacional. Além disso, os problemas de agrupamento geralmente possuem valores de  $a$  e  $b$  superiores ao mostrado no exemplo, logo a quantidade de soluções possíveis aumentaria significativamente a cada unidade de  $a$  e  $b$  aumentada (escala exponencial para aumento de elementos ( $a$ ) e grupos ( $b$ )). Esse rápido crescimento é denominado explosão combinatória.

$$N(a, b) = \frac{1}{b} \sum_{i=0}^b (-1)^i \binom{b}{i} (b-i)^a \quad (2.2)$$

É perceptível que encontrar a solução ótima, a partir da enumeração explícita das soluções, pode ser uma tarefa inviável. Além disso, muitos destes problemas pertencem ao conjunto de problemas denominados NP-difíceis (WOLSEY, 1998). Para tais problemas, não é conhecido nenhum algoritmo determinístico capaz de encontrar a solução ótima com complexidade de tempo polinomial.

De forma geral, estes problemas são tratados atualmente na literatura por duas diferentes metodologias, os chamados algoritmos exatos e heurísticos. Os métodos exatos são capazes, pelo menos na teoria, de fornecer soluções ótimas, isto é, uma solução viável que otimize (Maximize ou Minimize) o valor de uma função objetivo. Por outro lado, os métodos heurísticos fornecem uma solução viável sem garantia de otimalidade. Vale ressaltar que nos últimos anos observa-se uma iteração entre essas duas vertentes, com o objetivo de aproveitar o que há de melhor em cada uma.

### 2.1.1 Métodos Exatos

Os algoritmos exatos garantem a obtenção a solução ótima para problemas combinatórios. Geralmente, estes algoritmos definem estratégias iterativas que resolvem subpro-

blemas obtidos a partir de relaxações do problema original, limitando o espaço de busca a cada iteração, até que a solução ótima seja encontrada(WOLSEY, 1998).

Em alguns casos é possível encontrar algoritmos exatos "eficientes" para resolver problemas de otimização combinatória. Por exemplo, o problema de encontrar os caminhos mais curtos em um grafo, sob algumas suposições frequentemente encontradas na prática, pode ser resolvido pelos algoritmos Dijkstra ou Bellman-Ford. Estes algoritmos são capazes de fornecer soluções ótimas em tempos de execução de acordo com uma função polinomial de pequeno grau. Porém, para outros problemas, que podemos chamar de mais complexos, algoritmos que façam a busca em tempo polinomial ainda não são conhecidos. Isso quer dizer que a busca de uma solução ótima, para alguns problemas desta classe, pode demandar um tempo razoavelmente grande.

Para estes problemas existem, pelo menos, três abordagens diferentes e de propósito geral, ou seja, podem ser usados para resolver diferentes problemas: Algoritmos de planos de corte baseados em combinações poliédricas; Abordagens Enumerativas; e técnicas de relaxamento e decomposição (WOLSEY, 1998).

De forma geral, os métodos de planos de corte consistem em um método de busca iterativa que busca refinar um conjunto viável ou função objetivo por meio de inequações lineares, chamadas cortes. Tais procedimentos são utilizados para encontrar soluções válidas para os problemas(JUNGUER; REINELT; THIENEL, 1995). Geralmente esta técnica é combinada com outros tipos de técnicas, tais como as de enumeração e decomposição, dando origem a outros tipos de algoritmos.

As técnicas enumerativas buscam explorar o espaço de soluções de forma eficiente, enumerando-as de forma implícita. Devido ao fato do número de soluções para problemas de otimização combinatória ser significativamente grande, a enumeração deve ser implícita pois, do contrário, o algoritmo seria de pouca utilidade prática, equivalendo-se à resolução do problema pela força bruta.

Os métodos de decomposição, de forma geral, são utilizados para resolver um problema através da resolução de uma sequência de problemas de menor dimensão. Tais métodos também são utilizados para desenvolver algoritmos heurísticos. Entre estes métodos, um que tem se destacado para resolução de problemas baseados em redes E-R é a decomposição de Benders (BENDERS, 1962). Esta técnica não é limitada a problemas de programação linear inteira, podendo também ser utilizada para solução de problemas de programação não linear inteira mista.

Informações mais detalhadas de tais algoritmos, bem como informações sobre outras metodologias podem ser encontradas em(YOUNG, 1965; NEMHAUSER; WOLSEY, 1988; WOLSEY, 1998; HOFFMAN; PADBERG, 1999; PUCHINGER; RAIDL, 2005; GENOVA; GULIASHKI, 2011). Vale ressaltar que algoritmos exatos, apesar de garantirem a solução

ótima, podem demandar tempo impraticável com os atuais recursos computacionais, tornando-os inviáveis para problemas mais difíceis e de grande porte.

### 2.1.2 Heurísticas

O termo Heurística é derivado da palavra grega *heuriskein*, que significa encontrar ou descobrir. Ele é usado no campo da otimização para caracterizar um tipo de método de resolução de problemas. Formalmente, podemos definir uma heurística como uma técnica específica para um problema de otimização, ou classe de problemas de otimização, utilizada no projeto de algoritmos, para os quais não é possível garantir, ou não é conhecida nenhuma garantia, da qualidade das soluções geradas por esses algoritmos.

Diferentemente dos algoritmos exatos, vistos na seção anterior, que garantem encontrar a solução ótima, e de algoritmos aproximados que dão uma garantia de qualidade para o pior caso, os algoritmos heurísticos não dão nenhuma garantia da qualidade da solução por eles gerada (MARTI; REINELT, 2011).

Técnicas heurísticas têm sua devida importância devido ao fato que, em situações práticas, um engenheiro, um analista ou um gerente precisa tomar decisões o mais rapidamente possível para alcançar resultados desejáveis. Assim sendo, nestas situações onde a velocidade do processo é tão importante quanto a solução obtida, estes métodos tem se sobressaído. Isto porque bons métodos heurísticos geralmente obtêm uma boa solução com pouco esforço computacional.

Além do disso, existem outras razões para o uso de métodos heurísticos, entre os quais podemos destacar:

- Não se conhece nenhum método para encontrar a solução ótima para o problema.
- Embora exista um método para encontrar a solução ótima, por questões de hardware não é possível utilizá-lo.
- Por ser mais flexível, é possível modelar condições complexas mais facilmente em comparação aos métodos exatos.
- Pode ser incorporado a métodos exatos para alcançar a solução ótima.

Atualmente, na literatura, é possível encontrar diferentes classificações para os métodos heurísticos, cada uma utilizando diferentes abordagens. Neste trabalho, para um entendimento fácil e conciso, vamos dividir em três diferentes grupos: Heurísticas construtivas, Heurísticas de Busca Local e Meta-heurísticas. As descrições destes métodos, mostrados neste texto, são encontradas em Marti e Reinelt (2011).

Heurísticas construtivas são algoritmos que constroem um solução para um problema passo a passo. Geralmente estas heurísticas partem de um conjunto vazio, adicionando elemento ao conjunto ou combinando subconjuntos até que uma solução completa e viável seja obtida. Normalmente são métodos determinísticos e tendem a basear-se na melhor escolha. Outra característica desse tipo de heurística é que geralmente são simples e são fortemente dependentes do problema que se propõem a resolver. As heurísticas construtivas são amplamente utilizadas na otimização combinatória para gerar soluções de partida. Geralmente, a estas soluções são empregadas heurísticas mais robustas, como as heurísticas de busca local, para refinamento da mesma.

Em contraste com as heurísticas construtivas anteriormente mencionada, as heurísticas de busca local começam a partir de uma solução para o problema e tentam melhorá-la progressivamente. Para melhor entender o funcionamento destes métodos, considere  $e_e$  uma solução para um problema. É necessário definir uma estrutura de vizinhança, ou seja, uma função  $V$ , a partir da qual podemos chegar a soluções pertencentes a  $\mathbb{E} = \{e_1, e_2, \dots, e_n\}$  por meio de uma modificação, geralmente chamada de movimento. A cada passo da busca local são realizados movimentos de uma solução para outra, pertencente a  $\mathbb{E}$ , com o objetivo de encontrar uma solução de melhor qualidade. O método termina quando, para uma solução, não existe outra solução acessível a partir de  $V$  que melhore a função objetivo. Para esta solução se dá o nome de ótimo local. É importante ressaltar que uma função  $V$  gera um subconjunto de soluções pertencentes a  $\mathbb{E}$ .

As soluções que são geradas por  $V$  geralmente são exploradas de duas formas, conhecidas como: Primeiro de Melhora (do inglês First Improvement) e Melhor Aprimorante (do inglês KBS Improvement). Na primeira são verificadas as soluções até que se encontre uma solução que melhore a solução corrente. Na segunda, Melhor Aprimorante, todas as soluções geradas por  $V$  são verificadas, retornando a melhor entre elas. Podemos dizer que as soluções obtidas pelas heurísticas de busca local são dependentes de três recursos: da solução inicial, da função de vizinhança ( $V$ ) e do critério de exploração de soluções utilizados.

Apesar de permitir a obtenção de soluções melhores a partir de uma solução já conhecida, as heurísticas de busca local podem ficar presas em uma solução ótima local, que não necessariamente é a melhor solução para um determinado problema. A melhor solução para um problema é conhecida como ótimo global. Assim sendo, geralmente estes métodos são guiados por meta-heurísticas, estratégias de propósito geral para construção de heurísticas mais robustas. A Figura 3 exemplifica a ideia de ótimos locais e global para um problema de minimização. Enquanto para o ótimo local não existe regiões vizinhas com valores melhores de função objetivo, para o ótimo global não existe qualquer ponto, em todo o espaço, com valor melhor.

Meta-heurística é o termo utilizado para caracterizar algoritmos de propósito geral

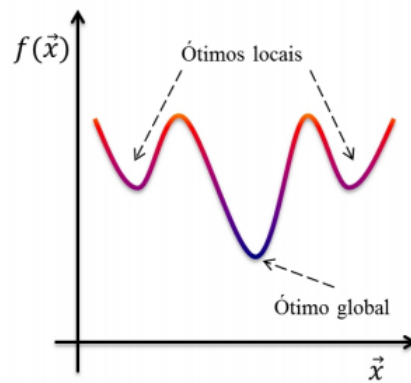


Figura 3 – Exemplos de ótimo local e ótimo global

que coordenam o uso de procedimentos de melhorias locais, como as heurísticas de busca local, juntamente com estratégias de alto nível para exploração do espaço de busca com o propósito de projetar heurísticas capazes de escapar de soluções ótimos locais e, ao mesmo tempo, realizar uma busca robusta pelo espaço de soluções.

Este tipo de método pode ser dividido em duas grandes classes: Meta-heurísticas baseadas em Busca Local e metaheurísticas baseadas em população (OSMAN, 2003). De forma geral, as metaheurísticas baseadas em busca local lidam com uma única solução por vez, mapeando diversas soluções ótimas locais através de algoritmos heurísticos de busca local ao longo de sua execução. Já as Meta-heurísticas baseadas em população, geralmente tem inspiração na natureza e lidam com um conjunto de soluções simultaneamente, gerando, a cada iteração, um novo conjunto de soluções. Estes algoritmos utilizam, de alguma forma, informações das soluções de conjuntos anteriores para explorar o espaço de soluções de forma inteligente.

Para cada uma destas classes existe uma grande quantidade de algoritmos diferentes. Um estudo mais detalhado destes algoritmos pode ser encontrado em diversos trabalhos, tais como Osman e Kelly (1996), Bianchi et al. (2009), Talbi (2009), Gendreau M. and Potvin (2010), Marti e Reinelt (2011), Boussaid, Lepagnot e Siarry (2013).

Embora os métodos heurísticos têm se mostrado eficientes, em relação a tempo e qualidade da solução, em muitas situações reais ainda é necessário utilizar recursos que os otimize. Isso porque quando se trata problemas de grande escala, ainda há falta de robustez e o tempo consumido por tais métodos, para obter soluções de qualidade aceitáveis, pode ser demasiado. Tais problemas surgem de demandas científicas ou industriais, e são chamados de problemas de grande escala devido ao fato do tamanho do espaço de busca ser extremamente grande. Assim, novas ideias têm sido propostas para reduzir essas limitações, tais como a incorporação de módulos de aprendizagem e paralelização.

O avanço no desenvolvimento de tecnologias para concepção de processadores (por exemplo, processadores *multicore*, arquiteturas dedicadas), redes ( LANs - como

Myrinet e Infiniband, redes de área ampla - WAN - como redes ópticas), e armazenamento distribuídos de dados tem tornado o uso da computação paralela cada vez mais popular. Outro aspecto muito importante é que hoje em dia, mesmo *laptops* e estações de trabalho são equipados com processadores *multicore*, que representam uma determinada classe de arquitetura paralela.

Além disso, a relação custo/desempenho está em constante diminuição. A proliferação de estações de trabalho poderosas e redes de comunicação rápidas proporcionaram o surgimento de *clusters* de processadores, redes de estações de trabalho e rede em larga escala de máquinas (grids) como plataformas de computação de alto desempenho. Estes recursos favorecem a concepção e implementação de meta-heurísticas paralelas.

Nesse contexto, usar metodologias que explorem o paralelismo, para contribuir na exploração no espaço de busca, passa ser uma evolução natural destes métodos. Ainda mais se considerarmos que as atuais linguagens de programação permitem que a paralelização seja utilizada de forma simples. A seguir será feita uma breve descrição sobre modelos de meta-heurísticas que utilizam este recurso.

## 2.2 Metaheurísticas Paralelas

Na computação paralela vários processamentos são feitos de forma simultânea em diferentes processadores ou núcleos. O paralelismo permite que uma decomposição da carga computacional seja feita, e posteriormente seja distribuída entre os processadores disponíveis. Em relação a decomposição, pode ser feita a nível de algoritmo, dados ou a estrutura do problema (CRAINIC; TOULOUSE, 2009).

A nível de algoritmo, denominado também de paralelismo funcional, diferentes tarefas, possivelmente trabalhando nos mesmos dados, são alocadas a diferentes processadores e rodam em paralelo, possivelmente trocando informações (não necessariamente). Já a nível de dados, é denotado paralelismo de dados ou decomposição de domínio, e refere-se ao caso em que o domínio do problema, ou o espaço de pesquisa associado, é decomposto e uma metodologia de solução particular é usada para resolver o problema em cada um dos componentes resultantes do espaço de pesquisa. O caso onde a decomposição é feita de acordo com estrutura do problema poderia ser realizada através de técnicas de programação matemática ou heurísticamente. Geralmente, algumas tarefas trabalham em sub-problemas correspondentes a conjuntos particulares de atributos (isto é, parte do espaço de pesquisa original), enquanto outras combinam soluções inteiras para o problema original.

Talbi (2009) cita alguns motivos para que o uso da computação paralela e distribuída seja usada na concepção e implementação de meta-heurísticas:

- Melhorar o tempo de busca: Um dos principais objetivos de paralelizar uma metaheurística é reduzir o tempo de busca. Isso ajuda a projetar métodos de otimização em tempo real. Este é um aspecto muito importante para alguma classe de problemas onde há requisitos rígidos no tempo de pesquisa, como em problemas de otimização dinâmica e problemas onde o tempo é crítico, como o planejamento em tempo real.
- Melhorar a qualidade das soluções obtidas: Alguns modelos paralelos de metaheurísticas permitem melhorar a qualidade da pesquisa. Isso é possível devido a troca de informações entre os processos de busca, que irá alterar seu comportamento no espaço de busca do problema. O objetivo principal de uma cooperação paralela em metaheurísticas é melhorar a qualidade das soluções. Neste caso, esta estratégia pode melhorar tanto convergência quanto o tempo de busca de soluções eficientes. Vale ressaltar que um modelo paralelo para meta-heurísticas pode ser mais eficaz do que uma metaheurística sequencial mesmo em um único processador.
- Melhorar a robustez: Uma metaheurística paralela pode ser mais robusta em termos de resolver de forma eficaz diferentes problemas de otimização e diferentes instâncias de um determinado problema. A robustez também pode ser medida em termos da sensibilidade da metaheurística aos seus parâmetros.
- Resolver problemas de grande escala: Metaheurísticas paralelas permitem resolver problemas de grande escala de maneira mais eficiente. Isso porque o uso correto do paralelismo proporciona uma pesquisa mais ampla no espaço de busca de forma mais rápida.

Uma quantidade significativa de estratégias para paralelização de metaheurísticas podem ser encontrados na literatura, como pode ser visto nos trabalhos apresentados por [Verhoeven e Aarts \(1995\)](#), [Cung et al. \(2002\)](#), [Grama e Kumar \(1999\)](#), [Alba e Nebro \(2005\)](#), [Crainic e Toulouse \(2009\)](#), [Talbi \(2009\)](#).

[Talbi \(2009\)](#) divide estas estratégias em três grandes modelos: Algoritmo, Iteração e Solução. Para o modelo baseado em Algoritmo, são utilizadas meta-heurísticas independentes ou cooperantes. Se as diferentes metaheurísticas forem independentes, a busca será equivalente à execução sequencial das meta-heurísticas em termos da qualidade das soluções. No entanto, o modelo cooperativo alterará o comportamento das metaheurísticas e possibilitará a melhoria da qualidade das soluções.

As metaheurísticas concebidas a partir de um modelo não cooperativo geralmente trabalham com diferentes métodos de busca, trabalhando de forma paralela. Tradicionalmente os métodos de busca são inicializados com diferentes soluções iniciais. Além disso, diferentes configurações de parâmetros podem ser usados com o objetivo de explorar melhor o espaço de busca. A concepção de metaheurística utilizando este modelo é simples. O

paradigma mestre/trabalhador é bem adaptado a este modelo. Um trabalhador, que pode ser um processo ou uma *thread*, implementa uma meta-heurística. O mestre, geralmente um processo, define os diferentes parâmetros a serem utilizados pelos trabalhadores e determina a melhor solução encontrada a partir das soluções obtidas pelos diferentes trabalhadores (Fig. 4).

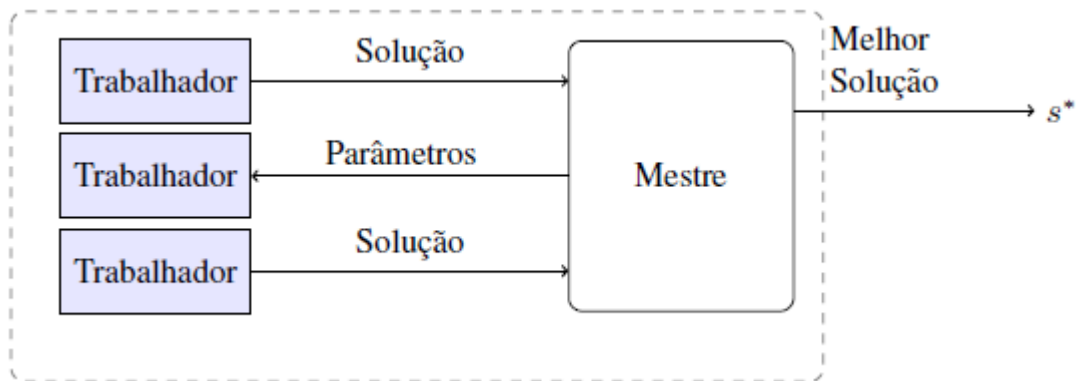


Figura 4 – Exemplo de modelo baseado em Algoritmo não cooperativo

Já as metaheurísticas concebidas a partir de um modelo cooperativo, os diferentes métodos de busca trabalham de forma paralela, trocando informações com o objetivo de obter soluções melhores. Neste tipo de modelo uma das questões levantadas, entre muitas, é quando os processos devem trocar informações. Pode acontecer de maneira cega, após um número pré definido de iterações, ou de acordo com um critério adaptativo. Outra questão importante é como a troca de mensagem será feita, em outras palavras, a topologia da troca de informação. Isto é importante porque o volume de troca de mensagens pode impactar no desempenho do algoritmo, bem como a topologia da rede de troca de mensagens/informações (DUHAMEL et al., 2013).

Estas trocas de informações geralmente são referenciadas como síncronas e assíncronas. No primeiro caso, todos os processos em questão param e se envolvem em alguma forma de comunicação e troca de informações em momentos determinados, coordenados por um processo principal. Já na comunicação assíncrona, cada processo é responsável por estabelecer comunicações com outras tarefas, de acordo com sua lógica interna.

Em relação a topologia, a ideia é indicar qual o fluxo das informações, em relação as tarefas que estão sendo executadas em paralelo. A Figura 5 ilustra as quatro principais topologias utilizadas (TALBI, 2009). Cada uma das topologias possuem características que devem ser levadas em consideração, por exemplo: grau máximo e diâmetro. Neste caso, quanto maior o grau dos nós, menos importante é o diâmetro do grafo, e mais intensiva é a informação trocada. Um pequeno diâmetro da topologia irá aumentar a informação trocada entre as buscas e, conseqüentemente, a probabilidade de convergência prematura. Vários esforços têm sido direcionados para estudar o impacto da topologia nos resultados

e comportamento das heurísticas, um estudo mais detalhado sobre este assunto pode ser encontrado em [Cohon et al. \(1987\)](#), [Belding \(1995\)](#), [Talbi \(2009\)](#).

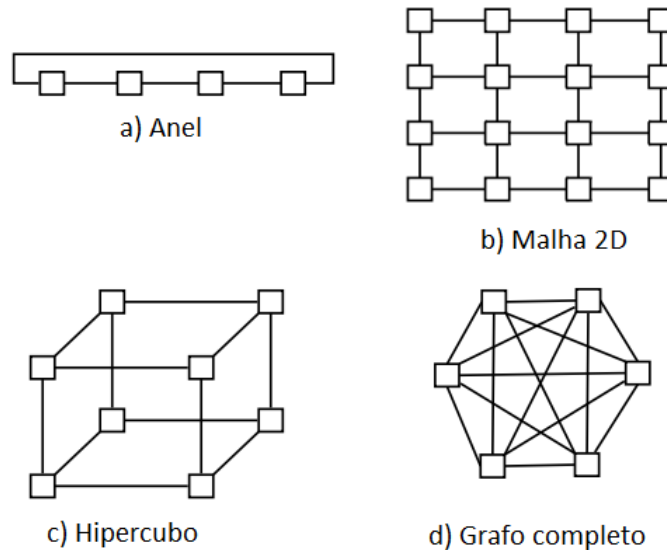


Figura 5 – Topologias clássicas para troca de informações entre tarefas/processos.

Já o modelo baseado em Iteração busca paralelizar cada iteração da metaheurística. Neste caso o comportamento da busca não é alterado, o objetivo é acelerar a busca em relação a tempo computacional. Esse modelo pode ser eficiente se olharmos que a exploração em grandes vizinhanças pode ser ineficiente de forma sequencial. Uma solução seria segmentar a vizinhança, e cada tarefa ficaria responsável em avaliar seu segmento. Outro caso em que este tipo de paralelização poderia ser aplicado são em problemas de otimização estocástica. Em problemas desta natureza várias avaliações de função são necessárias devido a incerteza dos dados. Isto ocorre por exemplo em funções objetivo não determinísticas, como é o procedimento de Monte Carlo, onde diferentes avaliações de função podem retornar diferentes valores. Geralmente, neste caso, é utilizada a média dos valores obtidos. Simulações poderiam ser executadas paralelamente com o objetivo de acelerar a resposta obtida.

No modelo baseado em Solução, assim como no modelo baseado em iteração, o objetivo é acelerar a busca. Porém neste caso, o que é paralelizado são os cálculos de função objetivo ou avaliação de restrições para uma determinada solução. Isto permite que funções objetivo mais complexas sejam avaliadas de forma mais eficiente. Neste caso o comportamento da meta-heurística também não é modificado. Os três modelos apresentados anteriormente podem ser utilizados em conjunto para compor uma meta-heurística, como pode ser visto na Figura 6 .

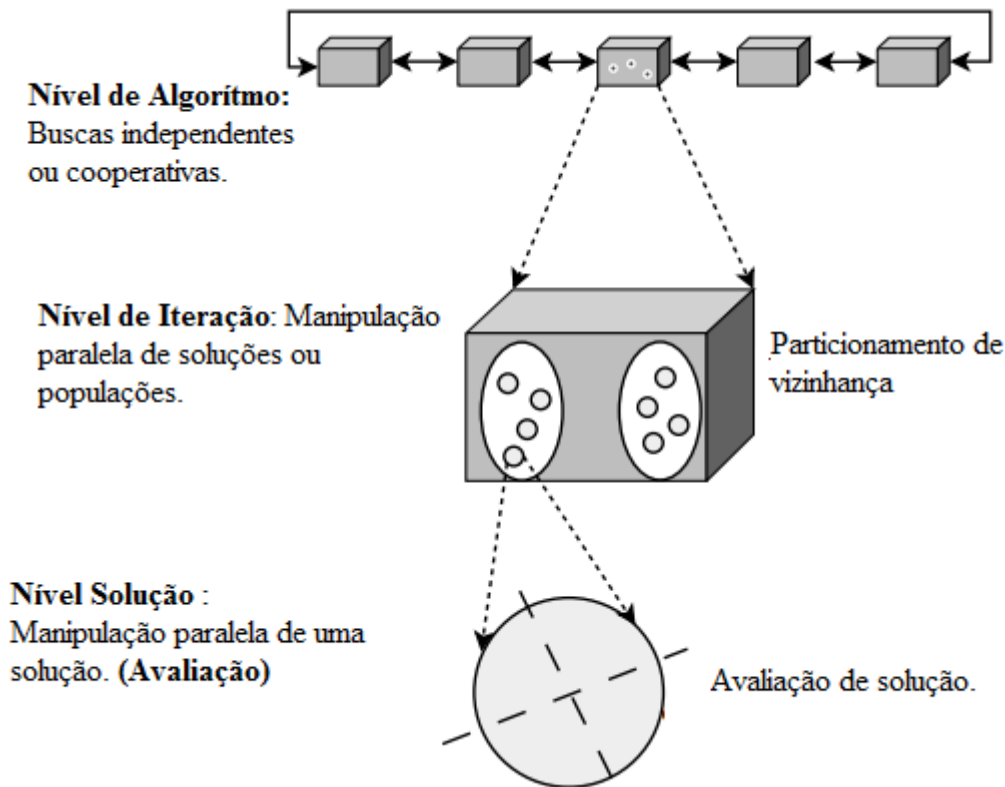


Figura 6 – Combinação de três modelos de meta-heurísticas paralelas. Fonte: Adaptado de (TALBI, 2009)

### 2.2.1 Arquitetura do Computador

Existem diferentes arquiteturas de computadores, no que se diz respeito a paralelismo, onde as meta-heurísticas podem ser implementadas. De forma simples, mas não suficiente para descrever as diferentes possibilidades de paralelismo nos dias atuais, Flynn (1972) propôs uma taxonomia baseada em duas características: sequência de instruções e sequência de dados. Essa taxonomia é uma das mais utilizadas para descrever, de forma geral, arquiteturas paralelas.

- SISD (*single instruction stream-single data stream*): Esta classe representa uma arquitetura composta por um único processador executando um programa sequencial. Aqui as máquinas possuem um único fluxo de instrução e um único fluxo de dados.
- SIMD (*single instruction stream-multiple data streams*): Essa classe representa arquiteturas paralelas onde o mesmo fluxo de instruções é executado em múltiplos fluxos de dados. Os processadores são restritos para executar o mesmo programa. Esta arquitetura é geralmente composta por processadores específicos. Tem um modelo de programação síncrono que se baseia na decomposição de dados. São muito eficientes na execução de algoritmos paralelos sincronizados que contêm cálculos regulares e transferências regulares de dados.

- MISD (*multiple instruction streams-single data stream*): Esta classe representa arquiteturas paralelas com múltiplos fluxos de instruções que são executados em um único fluxo de dados. Seria o caso de um pipeline de processadores aonde os dados vão sendo processados e passados para o processador seguinte. A proposta de implementação que mais se aproxima desta categoria é a da máquina de fluxo de dados. Na prática este tipo não existe.
- MIMD (*multiple instruction streams-multiple data streams*): Nesta arquitetura, múltiplos fluxos de instruções são executados em vários fluxos de dados. Os processadores podem executar diferentes instruções em diferentes dados. É o caso dos multiprocessadores, onde várias instruções podem ser executadas ao mesmo tempo em unidades de processamento diferentes controladas por unidades de controle independentes (uma para cada unidade de processamento). Permite execução de instruções diferentes para cada dado. Arquitetura utilizada em servidores com múltiplos processadores, sistemas multicomputadores (clusters) ou multi núcleos.

O foco deste trabalho é desenvolver heurísticas que possam ser executadas em máquinas do tipo MIMD. Embora todos os experimentos deste trabalho sejam executados em máquinas compostas com um ou mais processadores de vários núcleos, a ideia das heurísticas propostas podem ser extrapoladas para outras arquiteturas com as devidas adaptações. Por exemplo, quando se utilizam *clusters*, geralmente, os mecanismos pelos quais os processos paralelos são capazes de se comunicar, uns com os outros, é conhecido como memória distribuída, diferentemente do adotado neste trabalho, conhecido como memória compartilhada.

A memória compartilhada é um meio eficiente de transmitir dados entre processos. Em um modelo de memória compartilhada, os processos paralelos compartilham um espaço de endereço global onde é possível fazer leituras e escritas de forma assíncrona. O acesso simultâneo, assíncrono, pode levar a condições indesejadas e mecanismos como travas, semáforos e monitores podem ser usados para evitar este problema. Para este problema é dado o nome de região crítica, a Figura 7 ilustra este fato. É possível observar que quando um processo está sobre uma região crítica, qualquer outro processo, que precisa de um recurso protegido por alguns dos mecanismos citados anteriormente, precisa aguardar até que aquele recurso seja liberado pelo processo que está fazendo o uso. Isso pode ser observado na Figura 7, o processo B não realiza qualquer computação (está bloqueado) enquanto o processo A não libera o recurso que está sob uma região crítica. Os processadores multi-core convencionais suportam diretamente memória compartilhada, que muitas linguagens de programação paralelas e bibliotecas, como Cilk, OpenMP e Threading Building Blocks, são projetados para explorar.

Além disso, neste trabalho, chamamos de processos paralelos  $t$  threads trabalhando

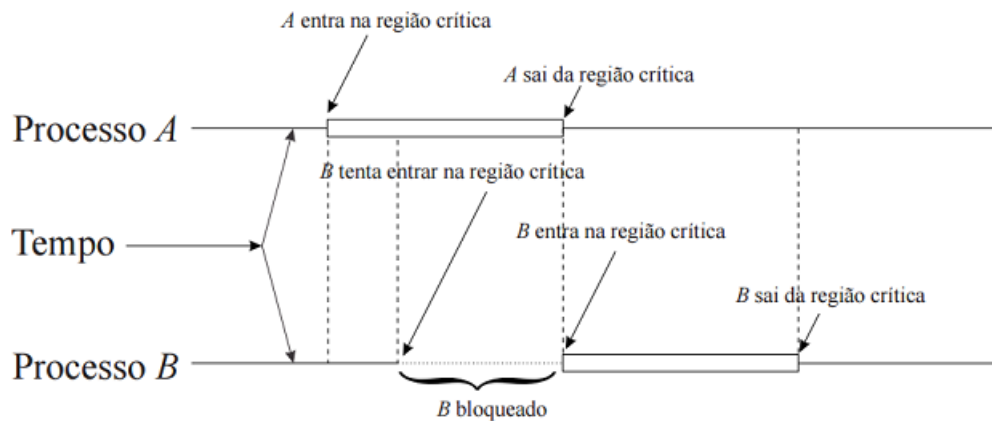


Figura 7 – Exemplo de escalonamento de processo em uma região crítica. Fonte: Adaptado de Tanenbaum (2008)

paralelamente. Há uma sutil diferença entre *threads* e processos na computação. Um processo é basicamente um programa em execução. Associado a cada processo está seu espaço de endereçamento, recursos, tais como registradores e arquivos, e outras informações relevantes para que um programa possa ser executado. De forma concisa, um processo pode ser entendido como um contêiner que armazena todas as informações necessárias para executar um programa. Antigamente, um processo possuía apenas uma única linha de execução (*thread*). Isto mudou com o surgimento de processadores de mais alto desempenho, que proporcionou uma nova unidade de processamento concorrente pudesse ser definida dentro do próprio processo, materializando novas unidades de fluxo de execução e assim pode-se ter múltiplos fluxos de execução (múltiplas *threads*) em um mesmo processo. O que as *threads* acrescentam ao modelo de processo é permitir que múltiplos fluxos de execução ocorram no mesmo ambiente do processo, com um grau de independência uma das outras mas compartilhando o mesmo espaço de endereçamento (TANENBAUM, 2008).

### 2.2.2 PThreads e OpenMP

Neste trabalho foram utilizadas duas interfaces de criação e manipulação de *threads*. A OpenMP, do inglês *Open Multi-Processing*, é uma interface para programação de algoritmos utilizando multi processamento com memória compartilhada. Este recurso está disponível para plataformas Unix e Windows. É constituída por um conjunto de diretivas de compilador, rotinas de biblioteca, e variáveis de ambiente que influenciam o comportamento do tempo de execução. Sua criação se deu na década de 90, quando fabricantes de máquinas com memória compartilhadas desenvolveram uma extensão do compilador Fortran para paralelizar alguns tipos de *loops*. Inicialmente, não foi muito utilizada devido ao grande interesse na época de se concentrar em máquinas com memórias distribuídas (*clusters*). No final dos anos 90, quando máquinas utilizando arquiteturas com memória compartilhada se tornaram mais comum, um novo padrão foi definido, e uma

nova interface para programadores C++ disponibilizada.

A importância do OpenMP se dá devido a falta de padrão encontrada até a sua proposta, ou seja, quem escrevia cada compilador inventava sua forma de criar e usar *threads*. Isso era uma coisa negativa porque caso mudasse de linguagem, sistema operacional, computador ou simplesmente compilador, era necessário aprender uma nova forma de implementar o paralelismo de acordo com os recursos disponíveis. Com a OpenMP foram definidas interfaces que especificam a forma de usar tal recurso, independente da linguagem, hardware, sistemas operacional ou compilador.

Atualmente existe uma organização, responsável em direcionar o desenvolvimento do OpenMP, que conta com diversas empresas do alto escalão de tecnologia em seu conselho. Algumas destas empresas são Hewlett-Packard, Intel, IBM, Sun e o departamento de Energia dos EUA. Para informações mais detalhadas sobre este recurso recomenda-se a leitura dos trabalhos apresentados por: [Chandra et al. \(2001\)](#), [Quim \(2003\)](#), [Chapman, Jost e Pas \(2008\)](#), [Mattson, Sanders e Massingill \(2013\)](#).

Já *PThreads* é uma biblioteca, para a linguagem C e C++, que provê uma interface que cria e manipula *threads* para o padrão POSIX. Este padrão é resultado de esforços para especificar um padrão para sistemas abertos, designada formalmente por IEEE 1003. O objetivo, assim como o OpenMP, foi padronizar o uso das *threads*. Quando esta biblioteca foi criada, seus desenvolvedores buscaram manter os princípios delineados pelos desenvolvedores de *kernel* de sistemas UNIX. Atualmente existe uma biblioteca de código aberto baseada neste padrão para sistemas Windows, porém sua implementação é diferente. Mais informações podem ser obtidas nos trabalhos apresentados por [Nichols, Buttlar e Farrell \(1996\)](#), [Cook \(2011\)](#).

## 2.3 Proposta (ILS-RVND-PR)

Neste trabalho, para os três problemas estudados, será utilizada a metaheurística, denomina aqui, ILS-RVND-PR. A abordagem consiste na combinação de vários métodos de busca trabalhando paralelamente de forma cooperativa. Os métodos comunicam, de forma assíncrona, somente para identificar a melhor solução encontrada e para combinar soluções no processo de intensificação, guiado pelo método *Path-Relinking* (PR). A estratégia assíncrona é a mais utilizada na literatura atualmente, e possui duas características que podem beneficiar o processo de busca. Primeiro, permite compartilhar informações sem que ocorram custos associados a sincronização dos processos, conhecido como *overhead*. Além disso, aumenta a adaptabilidade das metaheurísticas cooperativas, uma vez que sua capacidade de reagir e se adaptar dinamicamente à exploração do espaço de busca pelos cooperadores é significativamente aumentada em comparação com as outras estratégias de paralelização ([CRAINIC, 2015](#)).

A Figura 8 ilustra a heurística proposta, enquanto  $t$  threads executam as suas buscas utilizando o ILS-RVND, o *Path-Relinking* aplica um processo de intensificação a partir das soluções geradas por elas.

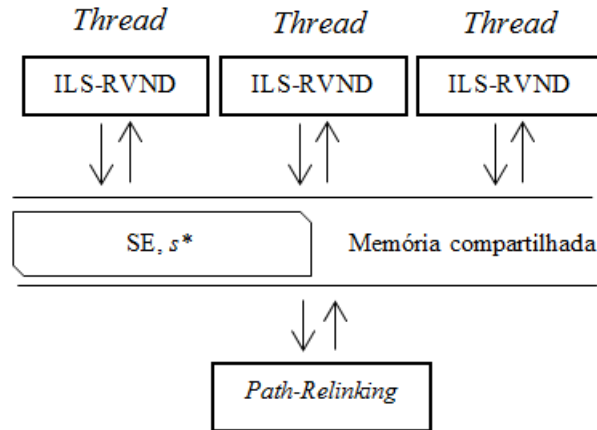


Figura 8 – Meta-heurística ILS-RVND-PR

Os métodos de busca combinam duas metaheurísticas muito conhecidas, *Iterated Local Search* (ILS) (LOURENCO; MARTIN; STUTZLE, 2010), e *Random Variable Neighborhood Descent* (RVND) (SOUZA et al., 2010), baseado no método *Variable Neighborhood Search* (VNS) proposto por Mladenovic e Hansen (1997).

O método ILS é uma metaheurística que explora o espaço de soluções por meio de uma sequência de perturbações a uma solução, seguidas de buscas locais. Este método é composto por um gerador de solução inicial, um procedimento de busca local, um procedimento de perturbação e um critério de aceitação. Inicialmente, uma solução  $s$  é gerada por um método de construção. Em seguida, é aplicado um método de busca local a  $s$  até que se alcance uma solução ótima local  $s'$ . Então, caso  $s'$  seja mais apta que  $s$  passará a ser a nova solução corrente. A esta solução é aplicada um procedimento de perturbação, e posteriormente aplicada novamente a busca local. Este processo se repete até que um critério de parada seja satisfeito.

RVND é um método de busca local que consiste em explorar o espaço de soluções por meio de trocas sistemáticas de estruturas de vizinhança, aceitando somente soluções de melhora da solução corrente e retornando à primeira estrutura quando uma solução melhor é encontrada. A cada chamada do método RVND a sequência de estrutura de vizinhança é alterada aleatoriamente. Isso reduz os riscos do algoritmo convergir prematuramente, além disso, é possível explorar diferentes áreas no espaço de busca, uma vez que a ordem das estruturas de vizinhança pode levar a diferentes soluções. O RVND foi usado como o método de busca local do método ILS. Por esse motivo, o método de busca deste trabalho foi denominado de ILS-RVND, como pode ser visto na Figura 8.

A combinação destes métodos tem sido utilizado com sucesso para resolver dife-

rentes problemas de otimização, tendo destaque em problemas de roteamento. Penna, Subramanian e Ochi (2013) utilizaram tal metodologia para resolver um problema de roteamento de veículos com frota heterogênea. Já Kramer, Subramanian e Penna (2015) utilizaram em um estudo de caso em uma indústria de bebidas, mais precisamente para resolver o problema de roteamento de veículos assimétrico com frota heterogênea limitada. Um versão paralela do método ILS-RVND é utilizado para tratar o problema de roteamento com coleta e entrega simultânea no trabalho apresentado por Subramanian et al. (2010). Enquanto Goncalves, Souza e Silva (2016) utiliza uma versão sequencial para tratar o mesmo problema. O problema de latência mínima, uma versão clássica do problema do caixeiro viajante, é tratado por diferentes versões desta metaheurística (SILVA et al., 2012). Outras aplicações podem ser vistas em: sequenciamento de tarefas (SANTOS; ARROYO, 2015); posicionamento de antenas (RUFINO et al., 2016); coloração robusta (SILVA; FROTA; SIMONETTI, 2015), entre outros.

Além de combinar tais métodos, a metaheurística proposta neste trabalho faz o uso de um mecanismo de intensificação conhecido como *Path-Relinking*. Este método consiste em explorar o caminho ou trajetória entre duas soluções, uma denominada solução guia, e outra guiada. Em cada iteração, movimentos são aplicados, estrategicamente, a solução guiada com o objetivo de obter a mesma estrutura da solução guia. A ideia principal deste método é que as soluções geradas, ao longo do caminho, compartilham um subconjunto significativo de atributos contidos nas soluções originais. Assim é possível que, ao longo deste processo, um novo ótimo local seja alcançado.

Os principais passos do método ILS-RVND são mostrados no Algoritmo 1. Inicialmente, uma solução é gerada através de um método de construção (linha 2), então  $t$ -threads são criadas para executar o restante do algoritmo. Para cada thread criada, o método RVND executa até que encontre uma solução ótima local (linha 6). Posteriormente, cada thread é responsável em inserir soluções em um conjunto, denominado soluções elite (SE), que servirá de repositório para o método *Path-Relinking* (linha 7). O conjunto SE possui tamanho  $t$ , e cada thread possui uma única posição onde a operação de escrita pode ser feita. Inicialmente são definidos dois critérios para atualizar uma solução no conjunto SE, qualidade e diversidade. Apenas soluções que possuem melhor valor de função objetivo e configurações diferentes são aceitas. Se após cinco iterações uma thread não atualiza SE, devido aos critérios estabelecidos, então a solução é atualizada apenas considerando o critério de diversidade. Posteriormente, é verificado se a solução atual  $s$  é melhor do que a melhor solução encontrada até o momento  $s^*$  (linha 8). Finalmente, o último passo do ILS-RVND consiste em escolher a próxima solução corrente ( $s$ ). Se a melhor solução encontrada foi atualizada, por qualquer uma das threads, ela se tornará a solução corrente. Caso contrário, uma perturbação é aplicada a  $s$ , passando a ser a nova solução corrente (linhas 9-13). Como critério de parada pode ser utilizado o tempo de execução ou números de iterações (linha 14). Paralelamente, o método *Path-Relinking* utiliza as soluções contidas

em SE para aplicar o seu procedimento de intensificação.

---

**Algoritmo 1: ILS-RVND**


---

```

1 início
2    $s \leftarrow$  Método de Construção
3   Inicializar SE
4   Criar  $t$  threads, cada uma deve executar as linhas 5-14
5   repita
6      $s \leftarrow$  RVND( $s$ )
7     Atualizar SE
8     Atualizar( $s, s^*$ )
9     Se existe uma nova  $s^*$  então
10       $s \leftarrow s^*$ 
11     senão
12       $s \leftarrow$  Perturbar( $s$ )
13     fim
14 até critério de parada seja satisfeito;
15 fim

```

---

Para o procedimento de perturbação, cada *thread* pode perturbar somente parte de uma solução. Isso favorece a exploração de diferentes regiões no espaço de busca. A Figura 9 ilustra um vetor representando uma solução para o problema em estudo, onde as partes hachuradas representam as posições disponíveis, para diferentes *threads*, para aplicar o procedimento de perturbação. Enquanto a primeira *thread* pode perturbar qualquer posição no vetor de solução, a última tem permissão para fazer modificações somente nas duas últimas posições.

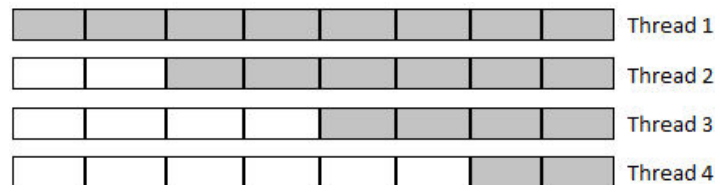
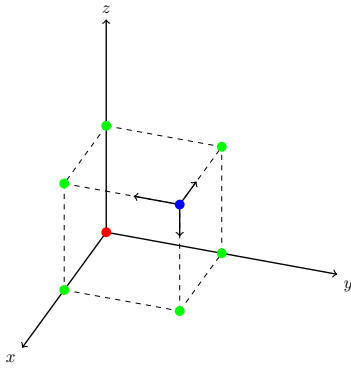
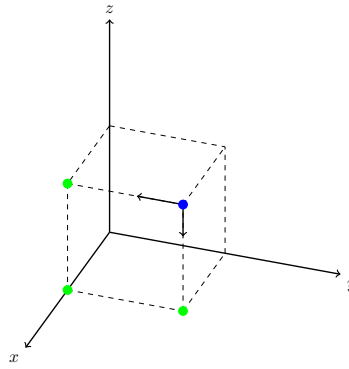
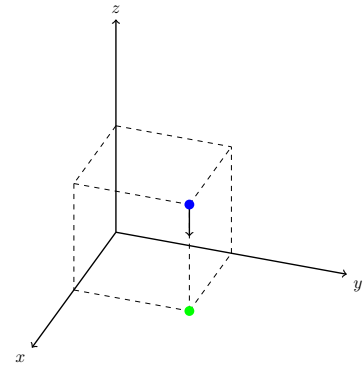


Figura 9 – Posições disponíveis para aplicar o método de perturbação por *thread*

Em todos os casos, a primeira *thread* criada tem permissão de modificar qualquer posição do vetor de solução. Para cada nova *thread* criada o número de posições disponíveis diminui de acordo com a equação (2.3), onde  $n$  é o tamanho do *array* que representa uma solução para o problema,  $t$  o número total de *threads* que serão criadas e  $t_c$  o número de *threads* que já foram criadas. Assim, para um problema com 8 nós e 4 *threads*, onde uma *thread* já tenha sido criada, o número de posições disponíveis para a segunda *thread* seria

Figura 10 – *thread 1*Figura 11 – *thread 2*Figura 12 – *thread 3*

6, ou  $E=2$ .

$$E = \lfloor \frac{n}{t} t_c \rfloor \quad (2.3)$$

As Figuras 10-12 ilustram uma situação onde dado um ponto de partida (azul), quais possíveis soluções (pontos verdes) poderiam ser alcançadas por cada *thread* criada. O ponto vermelho sinaliza uma solução inviável. Neste trabalho este ponto poderia sinalizar todos os nós como não concentradores.

Além disso, considere  $\pi_t$  sendo a porcentagem de posições alteradas, em relação as posições disponíveis, em uma iteração para uma determinada *thread*  $t$ . Considere também  $\pi_t^i$  e  $\pi_t^f$ , sendo, respectivamente, o valor inicial e máximo para este parâmetro. Inicialmente,  $\pi_t^i$  destas posições sofrem modificações. A medida que a melhor solução, entre todas as *threads*, não é melhorada, este fator de perturbação é aumentado, até que  $\pi_t^f$  das posições possam ser modificadas. Quando isso acontece, o valor da perturbação retorna ao valor inicial, ou seja,  $\pi_t = \pi_t^i$ . Caso uma solução melhor seja encontrada, o valor deste fator também retorna ao valor inicial. Além disso, a cada iteração, pelo menos uma posição de cada *thread* é alterada. Isso força a próxima solução, que passará pela busca local, a ser diferente da solução corrente.

O método ILS-RVND será explorado de duas formas. A primeira será chamada de ILS-RVND1, e consiste em várias buscas trabalhando em paralelo, e cada uma contendo um algoritmo sequencial. Neste caso, foi utilizado a interface PThreads para criação e manipulação das *threads* de busca. A segunda, será chamada de ILS-RVND2, além das buscas em paralelo, as estruturas de vizinhança também serão exploradas paralelamente. Aqui, na exploração das estruturas de vizinhança, foi utilizado a interface OpenMP, que segmenta o espaço de busca e distribui a carga de processamento entre as *threads* de maneira simples.

Embora a base dos algoritmos seja a mesma, o método ILS-RVND2 é paralelizado a nível de algoritmo, o que favorece a velocidade de convergência do mesmo. Porém, devido

ao fato das buscas serem paralelizadas, o número de *threads* de busca precisa ser menor para que não exista disputa de recursos entre elas, o que poderia gerar um comportamento muito ruim devido a espera para processamento. Por outro lado, o método ILS-RVND1 pode utilizar mais *threads* de busca, porém trabalhando de forma sequencial. Essa tática deixa o algoritmo mais lento quando comparado ao ILS-RVND2. Porém, devido a segmentação do espaço (perturbação) e um número maior de soluções que podem ser combinadas no PR, este algoritmo pode ser favorecido na exploração de espaços promissores.

## 2.4 Medindo o desempenho dos métodos heurísticos

Geralmente, o objetivo de uma heurística é encontrar uma boa solução em tempo razoável. Portanto, para verificar o desempenho destes métodos é natural que se utilize medidas que leve em consideração tanto a qualidade da solução quanto o tempo computacional, ou ainda, o esforço computacional. Além disso, devido a natureza estocástica deste métodos, uma série de experimentos independentes precisam ser feitos para que alguma conclusão possa ser observada.

Alguns indicadores podem ser usados para medir a eficiência de uma heurística, tais como número de vezes onde o ótimo foi obtido, número de avaliações de função ou tempo para alcançar uma solução alvo, desvio mínimo e médio em relação ao ótimo, etc. No caso onde o ótimo não é conhecido, ainda são utilizados a média/mediana dos valores obtidos de função objetivo, desvio de qualidade em relação a outros algoritmos, pior caso obtido, entre outros. No caso de heurísticas paralelas, existem vários indicadores, sendo um dos mais utilizados o *speedup*. O objetivo deste indicador é computar o ganho de desempenho de um método paralelo, em relação ao mesmo sequencial. Ao utilizar qualquer indicador é importante explicitar o objetivo, para que conclusões precipitadas não sejam obtidas.

Neste trabalho pretende-se responder algumas perguntas, tais como:

- Como as configurações de parâmetros afetam o tempo de convergência do algoritmo proposto?
- Existem gargalos computacionais nas heurísticas testadas?
- Como o tempo de execução é afetado pelos recursos disponíveis na máquina?
- Como a qualidade e tempo de execução do algoritmo se compara com os de seus principais concorrentes?
- Dada uma nova classe de instâncias, há mudanças significativas no comportamento dos algoritmos previamente estudados?
- Qual a qualidade das soluções em relação à melhor solução conhecida?

Assim sendo, serão utilizados os seguintes indicadores para nos auxiliar a responder estas perguntas:

- $P(\text{Alg1} < \text{Alg2})$ : informa a probabilidade de o algoritmo Alg1 encontrar uma solução mais rápida do que o Alg2. Inicialmente uma solução alvo é definida, posteriormente, para cada algoritmo, são amostrados 200 valores de tempo para alcançar uma solução alvo. Os resultados obtidos são exibidos usando o gráfico *Time-to-Target* (TTT), metodologia introduzida por [Alex, Resende e Ribeiro \(2007\)](#). Um gráfico TTT exibe no eixo das coordenadas a probabilidade do algoritmo obter a solução alvo dentro de um determinado tempo de execução (em segundos), mostrado no eixo das abscissas. A Figura 13 ilustra um gráfico TTT, quanto mais a esquerda uma linha, mais eficiente é o algoritmo. Quando as linhas se cruzam obtemos o valor do indicador. Neste exemplo, considerando Alg1 sendo a linha tracejada, e Alg2 a linha contínua, temos que  $P(\text{Alg2} \leq \text{Alg1}) \cong 0,9$ .

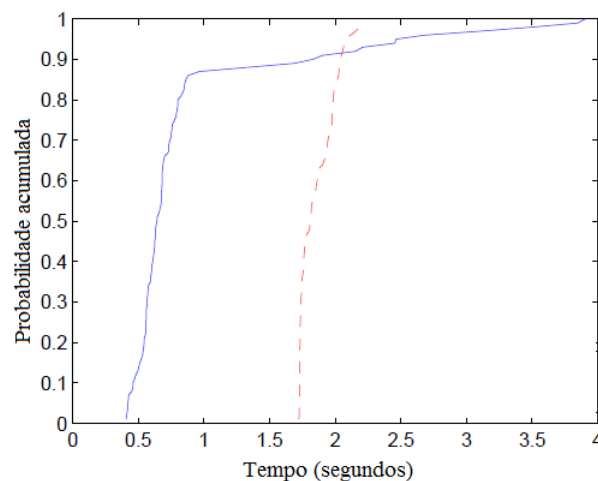


Figura 13 – Exemplo de um gráfico TTT

- $\#Best$  : contabiliza a quantidade de vezes que um algoritmo alcançou a solução ótima. Neste caso a solução ótima é conhecida. Quanto maior o valor deste indicador melhor é o algoritmo.
- KBS: também conhecido como *know best solution*, este indicador contabiliza a quantidade de vezes que um algoritmo alcançou a melhor solução, entre todos os algoritmos utilizados. Neste caso a solução ótima não é conhecida. Quanto maior o valor deste indicador melhor é o algoritmo.
- DevMed: contabiliza o desvio médio, em relação a função objetivo, entre a melhor solução computada, e as soluções obtidas por um determinado algoritmo (KBS). Considerando  $Q$  o conjunto de soluções utilizadas em um experimento,  $\phi_b^q$  sendo  $\#KBS$  ou KBS para um instância  $q \in Q$ , e  $\phi_i^q$  o valor de função objetivo obtida

na  $i$ -ésima execução de um algoritmo para uma instância  $q$ . Considere ainda  $x$  o número máximo de replicações. DevMed é calculado de acordo com a equação (2.4). Baixos valores para DevMed mostram consistência do algoritmo, pois indica que todas as soluções obtidas por ele estão próximas a  $\phi_b^q$ .

$$DevMed = \frac{\sum_{q=1}^{|Q|} \frac{\sum_{i=1}^x (\phi_i^q - \phi_b^q) / \phi_b^q}{x}}{|Q|} \quad (2.4)$$

- DevMin: contabiliza o desvio médio, em relação a função objetivo, entre a melhor solução computada, e a melhor solução obtida por um determinado algoritmo. A equação (2.5) mostra o cálculo desse indicador, onde  $\phi_{min}^q = \min_{i=1, \dots, x} \{\phi_i^q\}$ . Baixos valores para DevMin indicam que no melhor caso o algoritmo alcança soluções próximas de  $\phi_b^q$ .

$$DevMin = \frac{\sum_{q=1}^{|Q|} (\phi_{min}^q - \phi_b^q) / \phi_b^q}{|Q|} \quad (2.5)$$

- *Speedup*: este é um indicador que mensura a melhoria na velocidade de execução de uma tarefa executada em duas arquiteturas semelhantes com recursos diferentes. Neste trabalho, Speedup é definido como a razão  $T_s/T_p$ , onde  $T_s$  é o tempo médio tomado pelo algoritmo sequencial para encontrar a solução alvo em um conjunto de instância e  $T_p$  é o tempo requerido pelo algoritmo paralelo. Valores altos para *Speedup* indicam que o algoritmo paralelo é mais eficiente para encontrar a solução alvo.
- Média ( $\bar{\phi}^q$ ): contabiliza o valor médio obtido por um algoritmo após  $x$  rodadas para uma instância  $q$ .

$$\bar{\phi}^q = \frac{\sum_i^x \phi_i^q}{x} \quad (2.6)$$

Embora os resultados dos indicadores forneçam uma boa ideia sobre o comportamento dos algoritmos, não podem ser utilizados para obter conclusões mais gerais. Assim sendo, para complementar a avaliação de desempenho entre os algoritmos estudados, será realizado testes estatísticos com o objetivo de verificar se existe diferença estatística significativa entre eles. Com o objetivo de garantir condições experimentais homogêneas e independência dos erros, evitando a introdução de efeitos espúrios e tendenciosidades nas amostras, para todo experimento serão consideradas as seguintes premissas: testes executados em uma mesma máquina, de maneira aleatorizada, tanto em relação a instância quanto ao algoritmo; utilização da mesma semente geradora de números aleatórios em cada replicação de cada instância; tempo de execução igual para todos os algoritmos (MONTGOMERY; RUNGER, 2003).

Os experimentos serão projetados como um teste comparativo simples entre os algoritmos estudados. Para eliminar os efeitos das diferentes instâncias no desempenho dos métodos, cada uma delas foi tratada como bloco. O modelo estatístico que descreve os dados dos experimentos é dado de acordo com a equação (2.7).

$$\phi_{ijk} = \mu_i + \beta_j + \epsilon_{ijk} \quad (2.7)$$

, em que  $\phi_{ijk}$  é o valor da métrica calculada para o  $i$ -ésimo algoritmo, na  $j$ -ésima instância da  $k$ -ésima replicação.  $\mu_i$  é a média real da métrica para o  $i$ -ésimo algoritmo.  $\beta_j$  é o efeito no valor da métrica devido a  $j$ -ésima instância.  $\epsilon_{ijk}$  é a componente de erro aleatório. Ainda é possível decompor  $\mu_i$  em  $\mu + \tau_i$ , que representam, respectivamente, a média geral e o efeito do algoritmo.

Para testar as hipóteses nulas de ausência de diferença no desempenho dos algoritmos contra a hipótese alternativa de que pelo menos um algoritmo apresenta diferença em relação a um outro, será usado o teste de Friedman (FRIEDMAN, 1937). Os experimentos serão definidos da seguinte maneira:

- Hipóteses: H0 (hipótese nula), os algoritmos não apresentam diferença significativa quando aplicados ao problema estudado. H1 (hipótese alternativa), os algoritmos apresentem diferença significativa quando aplicados ao problema estudado;
- Fator: Algoritmo;
- Níveis: Algoritmos estudados;
- Replicações: 10;
- Região de rejeição H0: Consiste em todos os valores da estatística do teste utilizados, tais que sua probabilidade de ocorrência, sob H0, não supere o valor do nível de significância adotado nos testes.

## 2.5 Instâncias

Foram utilizados três conjuntos de instâncias para testar o desempenho dos métodos estudados, denominadas AP, BR e HL. As instâncias AP, também conhecidas como *Australian Post*, são amplamente utilizadas na literatura para problemas desta natureza, enquanto as instâncias BR e HL foram propostas neste trabalho. Os tamanhos das instâncias variam de 10 a 200 nós para as instâncias AP, e de 100 a 3038 nós para instâncias HL e BR. Foram considerados quatro diferentes valores para economia de escala,  $\alpha = \{0.2, 0.4, 0.6, 0.8\}$ . O conjunto de instâncias AP foi nomeado seguindo o padrão AP $n$ - $\alpha$ , onde  $n$  é o número de nós na rede e  $\alpha$  o fator utilizado para economia de escala. O conjunto de

instâncias propostas neste trabalho segue o padrão  $BRn-I-\alpha$ , onde  $I$  é um identificador de instância. Já o conjunto HL segue o padrão  $HLn_\alpha$ . A Figura 14 ilustra a disposição dos pontos de algumas das instâncias propostas neste trabalho.

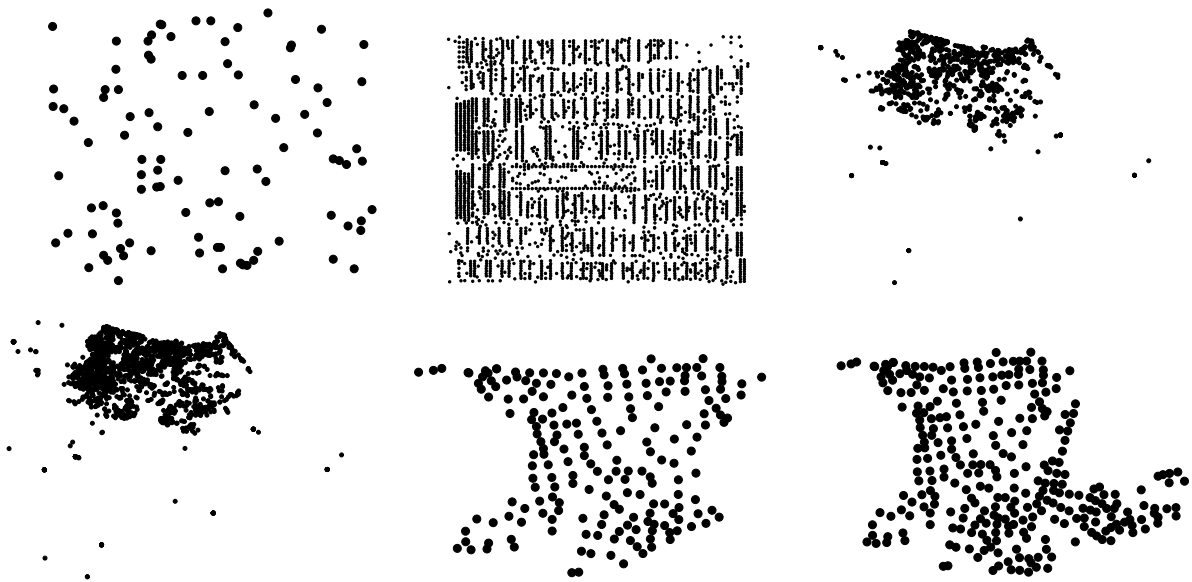


Figura 14 – Ilustração das instâncias BR propostas neste trabalho.

Os conjuntos de dados AP têm custos fixos apenas para instalação de concentradores para os primeiros 50 nós, assim, estes custos foram gerados para todas estas instâncias, usando uma distribuição Gaussiana com uma média igual a  $f_0$  e o coeficiente de variação ajustado para 40%; o objetivo foi simular como os diferentes custos de instalação variam em problemas reais. Observa-se que o coeficiente de variância é uma razão normalizada do desvio padrão para a média e mede a dispersão dos valores. Aqui  $f_0$  representa a diferença escalonada no valor da função objetivo entre um cenário em que um concentrador virtual está localizado no centro de massa e um cenário em que todos os nós são concentradores, como introduzido por Ebery et al. (2000). Os nós com fluxos mais elevados são selecionados para terem os custos fixos mais elevados, tornando a seleção de concentradores mais difícil. Camargo, Miranda Jr e Luna (2008) mostram como o esforço computacional é afetado quando o coeficiente de variação varia de 0% a 150%, sendo o maior esforço em torno do valor de 40%, em média, para esses conjuntos de dados.

Os custos fixos para as instâncias BR e HL também foram geradas baseada no método proposto por Ebery et al. (2000), e as demandas entre os pontos criadas aleatoriamente. A localização dos nós da instância BR foram baseadas nas instâncias para problemas de agrupamento propostas por Negreiros e Palhano (2006). Para as instâncias HL os nós foram distribuídos uniformemente no espaço. As instâncias e algoritmos utilizados neste trabalho podem ser encontradas no site [www.rclink.com.br/hub](http://www.rclink.com.br/hub).

## 2.6 Conclusão

Este capítulo teve como objetivo fornecer importantes conceitos para o entendimento deste trabalho. Inicialmente foi feita uma breve descrição sobre problemas de otimização combinatória, classe dos problemas abordados neste trabalho, bem como os principais métodos para resolvê-los. Além disso, introduzimos alguns conceitos sobre heurísticas paralelas, metodologia que será utilizada neste trabalho. De forma geral, foi mostrado a ideia da metaheurística que será utilizada para resolver as diferentes variantes dos problemas de rede baseadas em E-R. Isto foi feito com o objetivo de não repetir o mesmo conteúdo nos demais capítulos. Da mesma forma, foram apresentados os indicadores e instâncias que serão utilizados em todo o trabalho. No próximo capítulo será abordado uma revisão bibliográfica sobre problemas de localização de concentradores, com ênfase em redes do tipo E-R.

## 3 Revisão Bibliográfica - Redes baseadas em concentradores

Redes do tipo E-R (Eixo-Raio) têm sido utilizadas em diversos contextos e aplicações com sucesso, se tornando um importante campo de pesquisa que tem atraído a atenção de diversos pesquisadores. Se tratando de problemas de localização, este tipo de problema pode ser considerado um dos mais importantes. Neste capítulo é feita uma contextualização histórica em relação a problemas de localização, redes do tipo Eixo-Raio, bem como apresentados alguns exemplos de aplicações deste tipo de rede.

### 3.1 Problema de localização de Facilidades

O problema clássico de localização de facilidades tem o objetivo de localizar uma ou mais instalações, denominadas facilidades, para atender pontos de demanda a fim de minimizar ou maximizar uma função objetivo, que pode estar sujeita a algumas restrições, tais como capacidade, número máximo de facilidades, etc. A função objetivo varia de acordo com o contexto da aplicação: enquanto problemas aplicados a empresas do setor privado buscam minimizar custos ou maximizar lucro, nas do setor público o objetivo é maximizar os benefícios oferecidos ou minimizar os custos para oferecer um determinado serviço. Da mesma forma, o termo "facilidade", utilizado em problemas de localização, pode assumir diferentes significados dependendo do contexto da sua aplicação. No geral, esse termo refere-se a provedora de recursos, tais como hospitais, depósitos, escolas, entre outros. Já os pontos de demanda podem ser considerados os "clientes" da provedora de recursos, tais como empresas, pacientes, alunos, entre outros.

Problemas desta natureza são representados em dois tipos de ambientes: plano e rede. No caso do plano, as especificações de distâncias entre os pontos são dadas por métricas tais como distância Euclidiana, distância de Manhattan, entre outros. Já para o caso da rede, para computar a distância entre dois pontos são levados em consideração os possíveis caminhos entre cada par de pontos, eventualmente afetados por elementos topográficos, barreiras de diferentes tipos e orientação das conexões. Vale ressaltar que no último caso, em caráter aproximativo, pode ser usado a distância Euclidiana entre outras métricas.

Outra característica importante para este tipo de problema está relacionada a restrição envolvendo capacidade. Estes problemas podem ser classificados como sendo capacitados ou sem capacitação. Os problemas capacitados são aqueles em que as facilidades têm uma capacidade máxima para atender os pontos de demanda, enquanto os não

capacitados são aqueles que não têm restrições de capacidade para atender os pontos de demanda.

Segundo Reese (2005), este tipo de problema teve sua origem no século XVII, quando Pierre de Fermat procurou solucionar o seguinte problema: dados os vértices de um triângulo (três pontos no plano), encontrar o ponto (mediana), tal que a soma das distâncias entre cada um dos vértices em relação a mediana fosse o menor possível. Posteriormente, Hakimi (1965) generalizou este problema e é considerado o primeiro artigo na área de otimização motivado por conceitos parecidos ao problema de localização de concentradores. Variantes do problema estudado por (HAKIMI, 1965) foram aplicados nos mais diferentes contextos, como pode ser visto em: planejamento de redes secundárias de distribuição de energia elétrica (GARCIA et al., 2003); otimizar a designação de candidatos do vestibular aos locais de aplicação de prova mais próximos da residência do candidato (CORREA, 2000); alocação de unidades urbanas de lazer (BRONDANI et al., 2012); localização de escolas (PIZZOLATO; BARCELOS, 2004); instalação de serviços médicos de emergência (RUSLIN; GHANI, 2006); alocação de antenas de transmissão (ARROYO; MARQUES T. B. CORTES, 2006); localização de unidades de saúde 24 horas (ROSÁRIO; CARNIERI; STEINER, 2002), entre outros (DREZNER; HAMACHER, 1995). A formulação (3.1)-(3.5) representa a base teórica utilizada nestes problemas.

$$\text{Min } Z = \sum_{i \in I} \sum_{j \in J} d_{ij} a_{ij} \quad (3.1)$$

$$\sum_{j \in J} a_{ij} = 1 \quad \forall i \in I \quad (3.2)$$

$$a_{ij} \leq a_{jj} \quad \forall j \in J, i \in I \quad (3.3)$$

$$\sum_{j \in J} a_{jj} = p \quad \forall j \in J \quad (3.4)$$

$$a_{ij} \in \{0, 1\} \quad i \in I, j \in J \quad (3.5)$$

Onde:

- I: conjunto dos vértices considerados pontos de demanda.
- J: conjunto dos vértices candidatos a facilidades
- $p$  é o número de facilidades a serem localizadas.
- $d_{ij}$  a distância entre o ponto de demanda  $i$  e a facilidade  $j$ .

- $a_{ij}$  é uma variável de decisão, indicando a alocação da facilidade  $j$  ao ponto de demanda  $i$ . Sendo  $x_{ij} = 1$ , se o ponto de demanda  $i$  é atendido pela facilidade  $j$ , e  $a_{ij} = 0$  caso contrário.

A Função Objetivo é definida pela equação (3.1), esta tem o objetivo de minimizar o somatório da distância de cada vértice a mediana a que este foi associado. As restrições (3.2) determinam que um vértice deve estar associado a exatamente uma mediana. As restrições (3.3) determinam que um vértice pode ser associado a outro somente se este for uma mediana. As restrições (3.4) determinam o número máximo de vértices que podem assumir a função de facilidade. As restrições (3.5) impõem as condições de integralidade da variável de decisão. Este problema é conhecido como problema das  $p$ -medianas, a Figura 15 ilustra uma solução para este problema.



Figura 15 – Exemplo de solução para o problema de localização de facilidades

Como visto anteriormente, problemas de localização têm sido amplamente estudados pela comunidade acadêmica, se tornando uma importante linha de pesquisa em Otimização Combinatória. Um dos motivos para isso é que muitos problemas desta área, quando aplicados a situações reais, geralmente possuem natureza econômica, o que impulsiona tanto o desenvolvimento teórico quanto o prático no tema. [Pizzolato e Raupp \(2012\)](#) mostram que o desenvolvimento da pesquisa nesta área pode ser separado em três grupos, sendo eles:

- Trabalhos práticos: nessa vertente, estão contidos os trabalhos tipicamente assumidos por empresas de consultoria que buscam determinar a localização ideal de uma atividade industrial ou comercial específica. Alguns, entre vários, aspectos geralmente considerados nesses trabalhos de localização são: disponibilidade de mão de obra; acesso viário; regime pluviométrico; benefícios fiscais. Geralmente esses trabalhos são de natureza prática, e não costumam gerar publicações científicas, diante a interesses e sigilos envolvidos.

- Trabalhos acadêmicos: o objetivo nessa vertente é construir modelos matemáticos cada vez mais refinados. Estes modelos buscam contribuir de alguma forma com a comunidade científica, e não são necessariamente associados a problemas reais ou práticos. Os trabalhos acadêmicos resultam em avanços da ciência, e embora não se preocupem em associar seus métodos a problemas práticos, têm uma grande contribuição no aumento do elenco de problemas reais que podem ser resolvidos.
- Trabalhos aplicados: aqueles que fazem pesquisa visando resolver desafios do mundo real, notadamente complexos, beneficiando-se das contribuições proporcionadas pela área acadêmica. Diferentemente da vertente prática, aqui os resultados são compartilhados com a comunidade científica. Esse tipo de trabalho é usualmente considerado por pesquisadores em ciências aplicadas na busca por respostas a problemas observados pela sociedade e atividades industriais.

Todos estes grupos tem oferecido, de certa forma, suas contribuições, possibilitando que diferentes problemas de localização de facilidades, cada vez mais complexos, possam ser resolvidos de forma eficiente.

## 3.2 Redes do tipo Eixo-Raio

Apesar das semelhanças, os problemas do tipo Eixo-Raio distinguem-se dos problemas clássicos de localização. Nestes, geralmente, os pontos de demandas são atendidos, total ou parcialmente, por facilidades a serem localizadas. Por outro lado os problemas de redes E-R as demandas ocorrem entre pares de origem-destino. Nestes casos, a função dos concentradores não é o atendimento das demandas, mas o de funcionar como centros de comutação ou concentração de fluxo. Os concentradores permitem assim a consolidação, seguida de um redirecionamento e, posteriormente, de uma separação dos fluxos; sendo portanto facilidades intermediárias no caminho entre os diversos pares de origem-destino (CAMARGO, 2007).

Redes do tipo E-R permitem a diminuição do número total de ligações na rede, fazendo que os recursos sejam melhores aproveitados. Para conectar uma rede interligando todos os nós, a partir de uma rede ponto-a-ponto, são necessárias  $\frac{n(n-1)}{2}$  arestas. Por exemplo, em uma rede com 6 nós, completamente interligados, o número de ligações é 15. A mesma rede, configurada por uma estratégia do tipo E-R, com 2 concentradores, com apenas 5 ligações é possível interligar toda a rede, com os fluxos sendo canalizados através dos concentradores. A Figura 16 ilustra este caso, onde os triângulos representam os concentradores e os pontos os nós da rede.

Segundo O'Kelly e Muller (1994), este tipo de abordagem, redes E-R, permite a construção de uma rede onde um grande número de conexões diretas podem ser substituídas

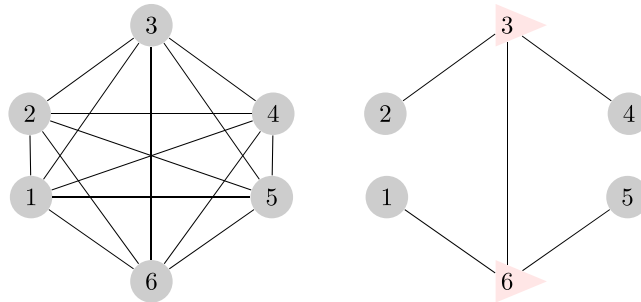


Figura 16 – Rede ponto-a-ponto x Rede baseado em concentradores

por menos indiretas, o que simplifica e reduz os custos de construção de uma rede, centralizando o manuseio e a triagem de fluxo, permitindo aos seus operadores aproveitarem as economias de escala por meio da consolidação dos fluxos.

A ideia das redes E-R iniciou com o trabalho proposto por [Goldman \(1969\)](#). Posteriormente essa ideia foi seguida por [O’Kelly \(1986b\)](#), [O’Kelly \(1986a\)](#) em estudos voltados para redes de transporte aéreo. Desde então, vários artigos têm sido publicados, com diferentes perspectivas e objetivos. [Farahani et al. \(2013\)](#) mostram que o foco dos trabalhos foi mudando com o passar dos anos; no final da década de 80 muitos esforços foram concentrados na formulação e modelagem do problema. Já na década de 90 grande parte dos trabalhos concentraram-se na criação de formulações e métodos de otimização. Nos dias atuais, grande parte dos esforços são para construir modelos e métodos avançados, que sejam capazes de resolverem problemas cada vez maiores de forma eficiente.

Em termos de modelagem matemática alguns trabalhos foram chaves para o desenvolvimento do tema. A primeira formulação para o problema de projeto de redes do tipo E-R foi proposta por [O’Kelly \(1987\)](#). Neste trabalho é proposta uma formulação quadrática, onde todos os nós são considerados candidatos à concentrador, o número de concentradores é conhecido previamente, não existe custo de instalação dos concentradores, a alocação dos nós não concentradores aos concentradores é simples e não existe restrição de capacidade nos concentradores. Posteriormente, [Campbell \(1994b\)](#), [Ernst e Krishnamoorthy \(1996\)](#) e [Ebery \(2001\)](#) propuseram modelos lineares para problemas correlatos. No trabalho apresentado por [Campbell \(1994b\)](#) foram propostas formulações para diferentes problemas baseados em concentradores, sendo que a primeira formulação linear para o problema proposto em [O’Kelly \(1987\)](#) é encontrada neste trabalho. [Aykin \(1995\)](#) propõem uma nova modelagem para o problema, e questiona a modelagem proposta por [Campbell \(1994b\)](#), indicando que a convenção, introduzida por [O’Kelly \(1987\)](#), de permitir no máximo dois concentradores ao rotear o fluxo entre dois pontos, nem sempre era atendida.

Outras proposta de formulações para redes do tipo E-R podem ser vistas nos trabalhos apresentados por [Camargo \(2007\)](#), [Alumur e Kara \(2008\)](#) e [Farahani et al. \(2013\)](#).

Uma das características importantes que diferem as formulações, inicialmente propostas, é em relação a alocação dos nós não concentradores. Existe basicamente duas possibilidades, alocação simples ou múltipla. Na alocação simples os nós não concentradores devem estar alocado a somente um concentrador, enquanto na alocação múltipla é possível que um não concentrador esteja alocado a mais de um concentrador. A Figura 17 apresenta duas redes baseada no conceito E-R, (a) com alocação simples e (b) com alocação múltipla.

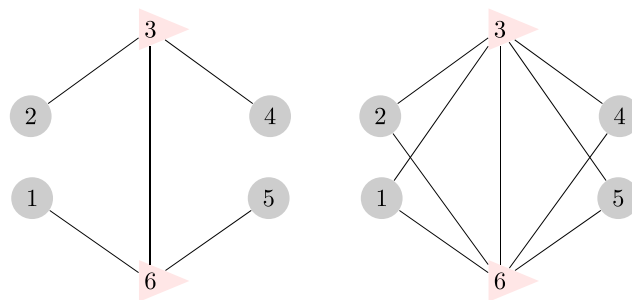


Figura 17 – Rede E-R com a) alocação simples e b) alocação múltipla

Não menos importante, outras características que influenciam na formulação dos problemas de redes baseadas em E-R são:

- custo de instalação: sem custo de instalação, geralmente neste caso o número de concentradores é conhecido a priori. Com custo de instalação, podendo ser fixo ou variável. Para os custos fixos, assim que um concentrador é instalado é estabelecido um custo fixo. Quando se trata de custo variável, geralmente o custo total de instalação se dá de acordo com o fluxo que passa no concentrador. Além disso, existem problemas onde é contabilizado o custo na instalação de arestas interligando os concentradores.
- número de concentradores: existe dois tipos de variantes, aquele em que o número de concentradores é previamente conhecido e outro onde é uma variável do problema.
- capacidade: o problema pode ser capacitado, neste caso existe uma capacidade máxima de fluxo que os concentradores podem intermediar. Caso contrário, quando não existe a restrição de capacidade, o problema é conhecido como não capacitado.
- domínio da solução: consideração de quais nós da rede são candidatos à concentrador. No caso de todos os nós serem candidatos a concentrador, o domínio é denominado

de Rede. Quando apenas um subconjunto de nós são candidatos, é denominado Discreto. Já no caso onde um concentrador pode ser localizado dentro de uma espaço pré definido, é denominado Contínuo.

- topologia: redes E-R podem seguir diferentes topologias, considerando tanto a rede formada entre os concentradores, quanto a rede formada com os nós não concentradores. Alguns exemplos são:
  - Estrela-Estrela: os concentradores estão interligados a partir de um concentrador local. Além disso, os nós não concentradores não tem ligação direta. A Figura 18 ilustra este tipo de rede, uma formulação para um problema baseado neste tipo de rede pode ser encontrado em [Yaman \(2008\)](#).

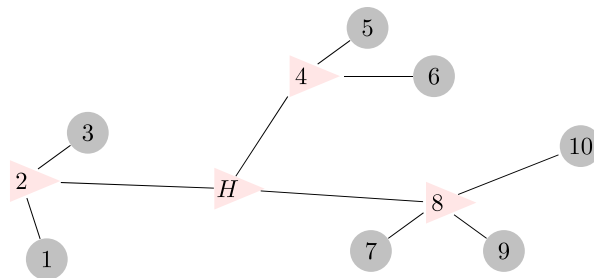


Figura 18 – Rede E-R Estrela-Estrela

- Anel-Estrela: os concentradores estão interligados em forma de anel, enquanto os nós não concentradores estão interligados aos concentradores em forma de estrela. A Figura 19 ilustra este tipo de rede. Uma formulação para um problema baseado neste tipo de rede pode ser encontrado em [Labbe et al. \(2004\)](#).
  - Anel-Anel: os concentradores estão interligados em forma de anel, e é necessário criar uma rota entre os nós não concentradores de forma que ela inicie e termine no concentrador. A Figura 20 ilustra este tipo de rede.
- Economia de escala: modo no qual a economia de escala é definida e utilizada. Inicialmente, é possível dividir em dois casos principais: onde a economia de escala é aplicada somente no transporte entre concentradores; onde a economia de escala também é aplicada no transporte entre concentradores e não concentradores, este desconto é mais comum aplicado ao processo de distribuição. Além disso, em relação ao valor da economia de escala, este pode ser constante, ou seja, não importa a quantidade de fluxo que passa no trecho, o desconto sempre é o mesmo. Outra

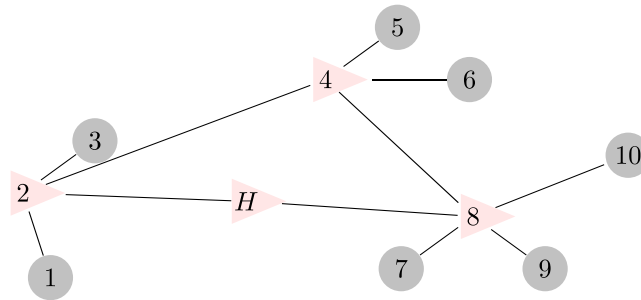


Figura 19 – Rede E-R Anel-Estrela

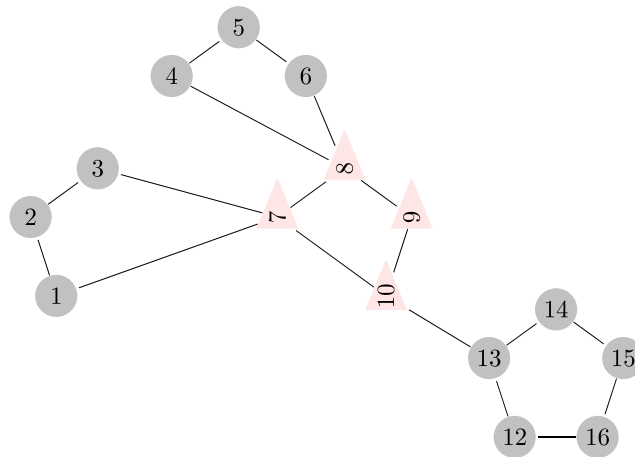


Figura 20 – Rede E-R Anel-Anel

abordagem é utilizar um fator que depende da quantidade de fluxo no trecho considerado, que varia, por exemplo, de acordo com uma função quadrática.

Vale ressaltar que estes são apenas alguns exemplos de características que envolvem redes do tipo E-R. Não temos o objetivo de enumerar todas as possibilidades, conhecidas na literatura, neste trabalho. Importante ressaltar que estas características influenciam diretamente na formulação e resultados obtidos.

Ainda, levando em consideração as formulações, é possível verificar importantes contribuições no decorrer dos anos. Novas variantes de problemas têm sido propostas, quer seja para aplicações específicas ou análises acadêmicas. Por exemplo, o trabalho apresentado por [Nickel, Schobel e Sonneborn \(2000\)](#) foi um dos primeiros a formular um problema de transporte público baseado em redes do tipo E-R. Posteriormente, outras formulações ([GERLACH, 2008](#); [CONTRERAS; FERNÁNDEZ; MARÍN, 2009](#); [CONTRERAS; FERNÁNDEZ; MARÍN, 2010](#); [SÁ; CAMARGO; MIRANDA, 2010](#)) foram propostas para problemas correlatos.

Formulações envolvendo incertezas são estudados por diversos autores. [Marianov e Serra \(2003a\)](#) consideram um problema de localização de concentradores, no contexto de transporte aéreo, considerando o possível congestionamento de aeronaves a partir de um sistemas de filas M/D/c. Já [Yang \(2009\)](#), no mesmo contexto, considera a incerteza na demanda entre os nós. [Alumur, Nickel e Gama \(2012\)](#) consideram incertezas no custo de instalação e na demanda entre os nós em um problema genérico. [Boukani, Moghaddam e Pishvaei \(2016\)](#) consideram incertezas nos custos de instalação, bem como na capacidade de cada concentrador.

Uma abordagem diferente, das apresentadas na literatura, para o problema de localização de concentradores capacitados com alocação simples é feita por [Costa, Captivo e Climaco \(2008\)](#). Ao invés de trabalhar com restrições referentes a capacidade de fluxo recebidas pelos concentradores, estes autores introduziram uma função objetivo ao modelo com o objetivo de avaliar o tempo de processamento dos fluxos de entrada nos concentradores. Deste modo, o problema foi formulado como um problema multi-objetivo, sendo um dos objetivos a otimização do tempo de serviço nos concentradores.

[Teymourian, Sadeghi e Taghipourian \(2011\)](#) propuseram uma formulação para definição de concentradores virtuais. Este problema consiste em localizar concentradores auxiliares que devem ser ativados em situações onde um concentrador da rede esteja sem possibilidade de fornecer serviços. Neste caso, um concentrador virtual pode ser considerado como um concentrador alternativo, predeterminado, que atua como um concentrador original em situações de emergência. O objetivo é que não ocorra a interrupção do serviço em toda a rede quando houver algum tipo de falha pontual na rede.

Formulações para o problema de localização de concentradores com demandas sensíveis ao preço são propostas por [O’Kelly et al. \(2015\)](#). Um dos objetivos do trabalho foi analisar a sensibilidade da demanda à qualidade de serviço. Além disso, as simulações computacionais possibilitaram a simulação da escolha do consumidor entre serviços concorrentes. Este trabalho contribuiu para uma melhor compreensão do tráfego nos concentradores com diferentes níveis de serviço, bem como o equilíbrio de preços em mercados competitivos.

[Real \(2015\)](#), propõe uma formulação, considerando características mais realísticas do que aquelas encontradas na literatura, para um problema de alocação de concentradores multi-nível. Tal trabalho procurou analisar o fluxo de transporte tanto de uma perspectiva local (país), quanto global. [Azizi et al. \(2016\)](#) propuseram uma formulação para um problema onde existe a possibilidade dos concentradores ficarem inoperantes, o que pode gerar prejuízo para os pontos de demandas interligados por eles. Uma formulação multi-objetivo, levando em consideração fatores ambientais e econômicos, pode ser encontrada em [Zhalechian et al. \(2017\)](#). Diversos outros trabalhos foram propostos, ilustramos apenas algumas contribuições para mostrar que novos problemas e formulações estão surgindo,

deixando em aberto a área para novas pesquisas e contribuições. Outras contribuições podem ser encontradas nos trabalhos apresentados por [Alumur e Kara \(2008\)](#), [Campbell e O’Kelly \(2012\)](#), [Farahani et al. \(2013\)](#);

Para resolver estes, entre outros, problemas de localização de concentradores, uma grande variedade de algoritmos têm sido utilizados. [Alumur e Kara \(2008\)](#) e [Farahani et al. \(2013\)](#) fazem uma ampla revisão da literatura, listando os principais métodos empregados entre os anos de 1987 e 2012. Um estudo analisando os últimos 25 anos de pesquisa na área de localização de concentradores é feito por [Campbell e O’Kelly \(2012\)](#), onde também é feita uma análise em relação as metodologias, deficiências e áreas promissoras para pesquisa.

Entre os algoritmos exatos podemos destacar abordagens baseadas em *Branch-and-Bound* (B&B) e o método de Decomposição de Benders([BENDERS, 1962](#)). Em se tratando do método de decomposição de Benders, [Camargo, Miranda Jr e Luna \(2008\)](#) propuseram uma abordagem para o problema de localização de concentradores não capacitado com alocação múltipla, capaz de resolver problemas que até então não era possível por métodos exatos. Posteriormente, [Contreras, Cordeau e Laporte \(2011\)](#) propuseram uma versão capaz de resolver instâncias maiores, com aproximadamente o dobro do tamanho daquelas utilizadas na literatura. Uma versão híbrida deste mesmo método com *Outer-Approximation*([DURAN; GROSSMANN, 1986](#)) é proposto por [Camargo, G. e Ferreira \(2011\)](#) para resolver o problema de localização de concentradores com alocação simples e congestionamento. Esta proposta foi capaz de resolver problemas aproximadamente 8 vezes maiores dos que os resolvidos até a presente data. [Sá et al. \(2013\)](#) também propuseram uma versão estendida do método de Decomposição de Benders para um problema aplicado ao transporte público, baseado em concentradores. Os resultados mostraram que a metodologia é mais eficiente que o software CPLEX para resolver instâncias de até 100 nós. Este método também foi empregado para resolver o problema de alocação de concentradores com demanda de muitos-para-muitos e roteamento de veículos ([CAMARGO; MIRANDA; LØKKETANGEN, 2013](#)). O algoritmo foi capaz de resolver problemas com instâncias de até 100 nós.

Considerando métodos baseados em B&B, [Cánovas, Garcia e Marin \(2007\)](#) propuseram um algoritmo híbrido, unindo um algoritmo B&B e uma heurística. Esta proposta foi capaz de resolver instâncias de até 120 nós para o problema de alocação múltipla não capacitado. [Labbé, Yaman e Gourdin \(2005\)](#) apresentou um método baseado em *Branch-and-Cut* para resolver o problema não capacitado com alocação simples. [Thomadsen e Larsen \(2007\)](#) propõem o mesmo método para o projeto de redes interligadas em duas camadas. Um problema de localização de concentradores com roteamento entre os nós alocados aos mesmos concentradores foi resolvido por [Martin, Gonzalez e Yaman \(2015\)](#) através de um algoritmo *Branch-and-Cut*. Neste trabalho foi possível resolver instâncias de

até 50 nós. [Contreras, Tanash e Vidyarthi \(2016\)](#) resolveram um problema de alocação de concentradores com topologia anel-estrela. Neste trabalho foi possível resolver instâncias de até 100 nós com um algoritmo *Branch-and-Cut*.

Em se tratando de heurísticas, as primeiras foram propostas por [O’Kelly \(1987\)](#) para o problema conhecido como localização de  $p$ -concentradores medianos. Posteriormente, [Klincewicz \(1991\)](#) desenvolveu algumas outras heurísticas para este problema, onde foi relatado a deficiência da alocação baseada apenas na distância, pois ignora a quantidade de tráfego entre os nós. [Skorin-Kapov e Skorin-Kapov \(1994\)](#) analisaram diferentes heurísticas, e propuseram uma nova baseada em Busca Tabu. Os resultados demonstram claramente a necessidade de propor alocações de nós não concentradores baseadas não somente em distâncias, mas também nos fluxos entre os nós. Posteriormente, vários outros métodos heurísticos foram propostos para diferentes problemas desta natureza. Dentre eles, se destacam aquelas baseadas em Algoritmos Genéticos (AG) e Busca Tabu.

[Abdinnour-Helm \(1998a\)](#) propuseram uma heurística híbrida, composta por um AG e Busca Tabu, para o problema de alocação de concentradores com alocação simples e não capacitado. Os dados das instâncias foram baseados no fluxo de passageiros entre 25 cidades americanas em 1970. Vale ressaltar que este conjunto de instância foi utilizado por muitos outros trabalhos, se tornando uma referência para problemas desta natureza. Outros trabalhos baseados em AG para este problema foram propostos por [Topcuoglu et al. \(2005\)](#), [Cunha e Silva \(2007\)](#), [Kratka et al. \(2007\)](#), [Takano e Arai \(2008\)](#), [Filipovic et al. \(2009\)](#), [Naeem e Berman \(2010\)](#), [Maric, Stanimirovic e Stanojevic \(2013\)](#) e [Gomes et al. \(2013\)](#). Um algoritmo evolucionário, utilizando um operador baseado na técnica *Path-Relinking*, foi proposto para resolver o problema com alocação múltipla e restrições relacionadas a distribuição de fluxo em [Qu e Weng \(2009\)](#).

Métodos baseados em Busca Tabu têm sido amplamente empregados a diferentes problemas de redes E-R. [Klincewicz \(1992\)](#) foi um dos primeiros autores a apresentar um procedimento baseado neste método para resolver um problema desta natureza. Nesse trabalho foi possível resolver problemas com até 52 nós para o problema proposto em [O’Kelly \(1987\)](#). Posteriormente, [Skorin-Kapov e Skorin-Kapov \(1994\)](#) propuseram outra heurística de Busca Tabu para o mesmo problema. Seus resultados foram superiores comparados aquelas apresentados na literatura em relação a qualidade de solução, porém em termos de tempo computacional a heurística se mostrou ineficiente. [Chen \(2007\)](#) propôs um algoritmo híbrido, composto pelos métodos *Simulated Annealing* e Busca Tabu, obtendo os melhores resultados, em relação a tempo e qualidade de solução, quando comparada às heurísticas da literatura da época. Atualmente, uma das melhores heurística para o problema também é baseada neste método, e pode ser encontrada no trabalho de [Abyazi-Sani e Ghanbari \(2016\)](#).

Outras aplicações do método Busca Tabu para problemas de localização de concen-

tradores em redes E-R podem ser encontradas em: Pamuk e Sepil (2001) *p-hub center*; Klicewicz (2002) localização de concentradores com fator de desconto variável (FLOWLOC); Marianov e Serra (2003a) localização de concentradores considerando congestionamento; Yaman e Carello (2005), Corberán et al. (2016) localização de concentradores capacitados com custo fixo e nas arestas; Calik et al. (2009) localização de concentradores com alocação simples e restrições de limite de tempo; Ishfaq e Sox (2011) localização de concentradores com alocação múltipla e restrições relacionadas a requisitos de tempo de atendimento; Ishfaq e Sox (2012) localização de concentradores com um modelo de filas para medir impacto do congestionamento em concentrador.

Não menos importante, métodos para otimização contínua têm sido utilizados para localização de concentradores no espaço contínuo, ou mesmo no espaço discreto, com auxílio de procedimentos de discretização (aproximação), como podem ser vistos nos trabalhos apresentados por Daganzo (2005), Alzamora et al. (2013), Saberi e Mahmassani (2013).

### 3.3 Aplicações para redes E-R

Existem hoje na literatura uma quantidade razoavelmente grande de aplicações das redes do tipo E-R, destacando-se aquelas aplicadas a problemas de transporte aéreo, transporte público urbano, serviços postais, sistemas logísticos e telecomunicações.

#### 3.3.1 Transporte Aéreo

Em se tratando de transporte aéreo, esse tipo de rede foi muito utilizada no planejamento do transporte nos Estados Unidos, principalmente com o Ato de Desregulamentação do transporte aéreo de 1978, que permitiu as companhias aéreas escolherem suas rotas e tarifas que seriam praticadas. Estas mudanças tiveram efeitos profundos sobre muitos aspectos da operação aérea daquele país, particularmente no que se refere a tarifas, nível de serviço, qualidade e segurança. Mas talvez o mais importante foi que as empresas aéreas alteraram a estrutura de suas rotas, desenvolvendo redes baseadas em concentradores, do tipo E-R (SILVA, 2004). O impacto da utilização desse tipo de rede sobre a rentabilidade das companhias aéreas desse país foi estudado por Toh e Higgins (1985).

Quando se trata de transporte de passageiros, de acordo com Bryan e O’Kelly (1999), além do custo operacional e do desempenho do sistema, o conforto dos passageiros deve ser levado em consideração. Além do impacto do custo nos preços das tarifas, os impactos de decolagem e pouso dos aviões também devem ser considerados no projeto da rede. Devido a importância do projeto das redes de transporte nesse segmento, esse tema atraiu vários pesquisadores com o objetivo de estudar a localização ótima de concentradores.

Além disso, houve o crescimento das demandas de envio de pacotes durante a noite nos anos 80, o que resultou em mudanças na organização do transporte norte-americano. [Hall \(1989\)](#) analisou o impacto das restrições noturnas e dos fusos horários na configuração de uma rede de frete aéreo. Posteriormente, [O’Kelly e Lao \(1991\)](#) apresentaram um modelo de programação linear inteira para planejar uma rede baseado na discussão feita por [Hall \(1989\)](#). Neste trabalho, é assumido que a localização dos concentradores é conhecida, o modelo é usado para determinar quais cidades podem ser atendidas por caminhão ao invés de aviões.

[Shaw S \(1993\)](#) fez uma análise das redes E-R utilizadas pelas mais importantes companhias aéreas dos Estados Unidos. O objetivo foi verificar as características da rede levando em consideração as diferentes estratégias de planejamento adotados pelas companhias e verificar a usabilidade dos modelos da época na aplicação em problemas reais. Para esse autor, um estudo mais eficaz deveria ser feito para qualificar os nós aptos a se tornarem concentradores, além disso, os modelos não contemplavam importantes características para sua utilização, de forma eficiente, em um problema real.

[Dobson e Lederer \(1993\)](#) propõem uma metodologia para fazer a escolha de horários de voos e preços de rota, de forma competitiva, para companhias aéreas operando em um sistema baseado em redes E-R. A escolha das rotas e tarifas são feitas de forma que as companhias maximizem seus lucros, e são baseadas nas decisões dos concorrentes.

Um estudo comparativo entre diferentes modelos, retratando políticas de serviço diferentes, para planejamento desse tipo de rede foi feito por [Jaillet, Song e Yu \(1996\)](#). Nesse trabalho o número de concentradores entre origem e destino não foi determinado. Os autores chegaram a algumas conclusões importantes, tais como: os candidatos a concentradores dependem mais da sua localização do que seu nível de demanda; A política de apenas um parada entre origem e destino pode ser tão boa quanto a duas paradas, porém a primeira permite a companhia a ganhar uma parcela maior de mercado; Uma boa rede está longe de ser do tipo E-R tradicional.

Levando em consideração o conforto dos passageiros, alguns estudos, tais como os feitos por [Sasaki, Suzuki e Drezner \(1999\)](#) e [Kara e Tansel \(2000\)](#), consideram o projeto da rede E-R utilizando apenas um concentrador entre origem e destino. Ao fazer isso o problema pode ser transformado em um problema de p-medianas.

Outro aspecto importante observado pelas operadoras de transporte aéreo é o congestionamento. Isso pode conduzir a atrasos, perdas de ligações e desvios de bagagem, o que será interpretado pelo cliente como um serviço de baixa qualidade. No transporte aéreo o congestionamento pode ser abordado tanto nos aeroportos quanto nas aeronaves. Trabalhos com tal enfoque podem ser visto em [Marianov e Serra \(2003b\)](#), [Elhedhli e Hu \(2005\)](#) e [Camargo e Miranda \(2012\)](#).

Lin e Lin (2003) fizeram um estudo para projetar uma rede de transporte aéreo de cargas baseada em dados obtidos de uma empresa Asiática. O objetivo foi definir o tamanho da frota de aeronaves, rotas e trilhas de frete de modo que os custos fixos e operacionais fossem os mínimos, e que atendesse todas as restrições operacionais, tais como tempo de deslocamento, capacidade das aeronaves, etc. Não foram utilizadas as suposições de uma rede clássica baseada em E-R neste estudo. O problema foi modelado de forma que fosse possível consolidar cargas em diferentes pontos de carga/demanda antes de enviar para um concentrador. Além disso, a modelagem também permitiu consolidar cargas entre concentradores antes de alcançar o destino final. Foi observado que para o mercado em estudo, os centros, pontos de demanda, são próximos, o que proporciona a utilização deste tipo de rede de forma eficiente.

Um estudo sobre o sucesso e crescimento das companhias aéreas de baixo custo, nos Estados Unidos, entre os anos de 2002 e 2008 foi feito por Cook (2008). Este trabalho procurou comparar as características econômicas e operacionais dos sistemas de rotas ponto-a-ponto e E-R. Os autores ressaltam que embora as operadoras de baixo custo dos EUA sejam frequentemente caracterizados como sistemas operacionais ponto-a-ponto, na prática poucas realmente trabalham neste ambiente. Geralmente são encontrados modelos híbridos, que fazem uso de redes E-R.

Oktal e Ozger (2013) fizeram um estudo para localização de concentradores utilizando dados de empresas aéreas de carga da Turquia. Um modelo multi alocação foi proposto considerando distância máxima de vôo entre aeroportos e capacidade de um aeroporto para receber determinado tipo de aeronave. Os autores reforçam a ideia que a localização do concentrador é um fator importante a ser considerado. Além disso, as características das aeronaves devem ser levadas em consideração, como fatores significativos, que afetam o processo de seleção do concentrador, especialmente quando isso requer investimentos importantes nos aeroportos e incorre em custos operacionais.

Outros trabalhos com diferentes abordagens e objetivos podem ser vistos em Farahani et al. (2013).

### 3.3.2 Transporte público urbano

Atualmente temos que conviver com o problema do trânsito urbano em praticamente todos os centros importantes do mundo. O crescimento das áreas metropolitanas tem exigido dos governantes uma reestruturação e expansão das suas redes de transporte público com a finalidade de melhorar a mobilidade urbana e reduzir problemas de tráfego, tais como congestionamento, consumo de energia, poluição, entre outros (SÁ, 2015).

Nickel, Schobel e Sonneborn (2000) apresentaram um dos primeiros trabalhos utilizando redes baseadas E-R para o problema de transporte público. Nesse trabalho

foram considerados tanto o custo de se instalar um concentrador, quanto o custo de instalação de uma conexão entre eles. Além disso, o problema é modelado de forma que a economia de escala entre concentradores seja grande, o que dá prioridade aos usuários escolherem os caminhos de menor custo (valor monetário), e não de menor tempo ou distância.

Contreras, Fernández e Marín (2010) modelaram o problema onde uma rede de concentradores é conectada como uma estrutura de uma árvore. Potenciais aplicações surgem no desenho de sistemas de transporte público onde o custo de instalação da infraestrutura necessária domina consideravelmente os custos de roteamento. Deste modo, a completa interconexão dos concentradores é irrealística. Outro problema considerando a mesma dificuldade é estudado por Sá et al. (2013). Nesse trabalho, o objetivo foi localizar um conjunto de concentradores conectados por meio de um trajeto simples (ou linha) procurando minimizar o tempo total de viagem ponderado pela demanda entre os pares de origem e destino. Ainda é levado em consideração o tempo gasto para acessar e deixar a rede de concentradores.

Hosapujari e Verma (2013) propuseram um trabalho com o objetivo de encontrar um conjunto de rotas ótimas entre concentradores (estações) e as rotas de alimentação (bairros), no sistema de transporte público utilizando ônibus. O modelo utilizado levou em consideração os custos do passageiro e operador, levando em consideração fatores como lotação máxima para cuidar do interesse do passageiro, e lotação mínima para cuidar do interesse do operador. Além disso, também é considerado o tamanho da frota disponível. Quando comparado com um sistema ponto a ponto, os resultados mostraram que é possível reduzir a frota sem aumentar significativamente o tempo de viagem. Por outro lado, um pequeno aumento na tarifa dos usuários seria necessário para tornar o sistema viável.

Um estudo utilizando dados de Dalian, cidade localizada na China, para localizar concentradores no sistema público de transporte foi feito por Yu et al. (2013). Diferentemente de outros estudos, onde os candidatos a concentradores são definidos a partir do conhecimento do problema, nesse trabalho foi utilizado um critério para defini-los. A escolha é baseada em duas características de um nó: acessibilidade e conectividade. Acessibilidade do nó indica a dificuldade de chegar em outros nós a partir dele. Já a conectividade indica a dificuldade de chegar nos principais destinos urbanos (Metrô, Fábricas, etc) vindo deste nó. Segundo os autores, esta abordagem forneceu uma boa indicação para o governo tomar a decisão de onde instalar novas estações.

Como alternativa para otimizar o transporte público da cidade Zhoushan, na China, que teve seu tráfego urbano consideravelmente aumentado no decorrer dos anos, Yu et al. (2015) propôs um planejamento de rede marítima baseada em redes E-R para resolver o problema do congestionamento do tráfego urbano. Isso foi possível devido ao fato desta cidade estar localizada em um arquipélago, sendo que muitos pontos são interligados por

água. O objetivo do trabalho foi propor uma rede de forma que este serviço fosse mais abrangente, atendendo mais a população, uma vez que devido a pouca demanda entre os pontos, a oferta de translados era consideravelmente pequena. Além disso, o trabalho também teve como objetivo diminuir os custos do serviço, para que fosse possível oferecer mais translados, sem prejuízo para o operador.

Sá, Contreras e Cordeau (2015) propuseram uma solução para redes de transportes urbanos com o objetivo de criar redes onde o tempo de deslocamento entre os nós fosse o menor possível. Considerou-se que existe um meio de transporte rápido entre os concentradores, ou pelo menos mais rápido quando comparado com o tempo de deslocamento entre outros pontos da rede. Além disso, é considerado que existe diferentes linhas de transporte rápido interconectados. Os tempos relacionados a embarque e desembarque, quando existe mudança no tipo de transporte, é contabilizado. Segundo os autores, uma aplicação em potencial para este problema é nas localizações de estações em grandes centros urbanos, onde as diferentes linhas de transporte rápido podem ser representadas pelos *Bus Rapid Transit* (BRT) e metrô, enquanto os nós de demanda da rede podem ser considerados bairros ou estações de táxi.

Contreras, Tanash e Vidyarthi (2016) estudaram o caso onde ligar os concentradores em forma de anel é mais vantajoso. Em algumas situações, uma linha circular pode ser desejável não só devido a requisitos de confiabilidade, mas também porque oferece um caminho alternativo que pode reduzir consideravelmente o tempo de viagem entre os pontos de demanda.

### 3.3.3 Telecomunicação

Em telecomunicações, os fluxos a serem enviados são dados que devem ser entregues entre origem e destino. Existe um leque de aplicações, tais como videoconferência, redes de telefonia, processamento distribuído, entre outros. Nesse contexto, nós concentradores são os multiplexadores, *switches*, *gates*, entre outros. As ligações entre os concentradores podem ser diferentes tipos de mídia, tais como fibra ótica e cabo coaxial (GELAREH, 2008).

Gavish (1987) utiliza o conceito de redes do tipo eixo-raio em um sistema de computação distribuída, no qual cada ponto produz uma transação que é armazenada em um banco de dados. Um número de computadores deve ser instalado em locais diferentes para processar as transações e gerar relatórios a partir delas. Nesse problema os computadores são os concentradores, os pontos de origem são as fontes de transações e os pontos de destino os relatórios. O fluxo é modelado pela demanda de dados entre concentradores para gerar um determinado relatório.

Klincewicz (1998) apresenta um revisão da literatura para redes do tipo tributária

e *backbone* onde os concentradores são escolhidos a partir de um número discreto de pontos. A Figura 21 ilustra os principais tipos de rede tributárias conectadas a concentradores que formam uma rede do tipo *backbone*. Aos concentradores C e D existem duas redes conectadas, uma em árvore e outra em caminho, respectivamente. Esses dois tipos são frequentemente referenciados como linhas do tipo multi-ponto. Ao concentrador A existe uma rede em estrela, configuração que pode ser usada em telefonia. Uma rede em anel está conectada ao concentrador B, configuração comum em aplicações que fazem uso de fibra ótica (CAMARGO, 2007).

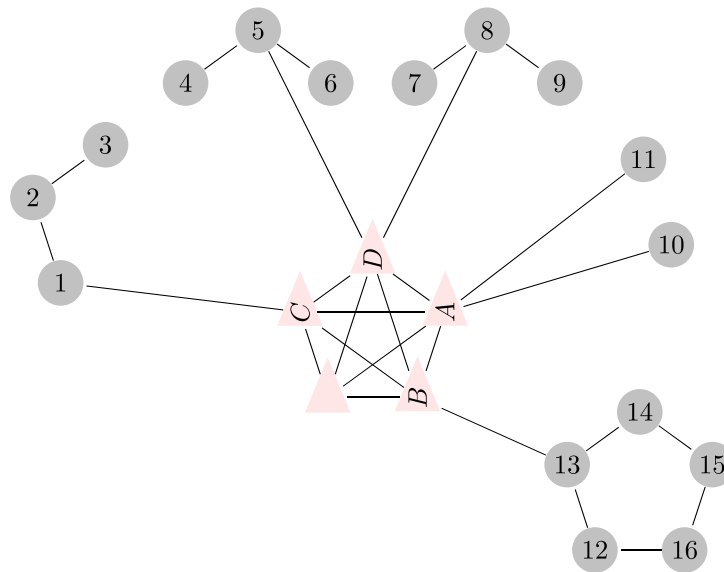


Figura 21 – Ilustração de uma rede *backbone*/tributaria.

Problema envolvendo topologia utilizada em redes privadas, MANs e sistemas de rede CATV são estudados em diversos trabalhos. Nesse problema os concentradores são ligados a outros concentradores até um concentrador central de forma hierárquica. Lee (1993) considera o problema onde existem diferentes tipos de concentradores, cada um com capacidade e custo diferentes. O problema requer não só a escolha das localizações dos concentradores, mas também a escolha de qual tipo deve ser inserido no local.

Pirkut e Nagarajan (1992) desconsideram a capacidade dos concentradores e força a sua rede de concentradores a formar uma rede do tipo árvore com o concentrador central, e caminho entre concentradores. Assim, com exceção do concentrador central, nenhum outro possui mais de duas ligações incidentes a ele. Este problema é uma variante do problema hierárquico de alocação de concentradores baseado em redes E-R.

Kim, Kim e Ryu (1996) apresentam um problema onde os nós tributários representam estações de base para serviços de comunicação pessoal (PCs) conectados em estrela a controladores, que por sua vez estão conectados em estrela a um concentrador central. Este modelo pode ser visto como tendo uma estrutura global de árvore / estrela. Como

alternativa, pode-se visualizar o modelo como uma estrela / estrela / rede em estrela de três níveis.

Carello et al. (2004) trabalharam em um problema de localização de concentradores baseado em redes E-R com dois tipos de nós: nós de acesso, que representando a fonte e destino de demandas de tráfego, mas não podem ser conectados diretamente, e nós de trânsito, que não possuem demanda de tráfego própria, mas coletam tráfego de nós de acesso e roteiam através da rede. Na rede estudada foi considerado três diferentes tipos tráfego: comutação de circuitos, comutação de pacotes e interligado. Isto exigiu diferentes estratégias de alocação para diferentes tipos de tráfego. Foram propostas diferentes heurísticas para resolver pequenos problemas, de até 49 nós. Os autores salientaram que parte do trabalho foi executado em parceria com uma empresa de telecomunicação da Itália.

Kim e O'Kelly (2009) consideram que concentradores são elementos críticos nas redes de telecomunicações e transporte, porque eles desempenham um papel vital no fluxo do tráfego. Assim, o projeto de redes mais confiáveis em sistemas E-R é um problema crítico, considerando que muitos *backbones* comerciais da Internet são bastante vulneráveis. Os autores propõem um novo problema de localização de concentradores, denominado "problema de localização p-concentradores confiável, onde o objetivo é maximizar o desempenho da rede em termos de confiabilidade. Redes de telecomunicações dos Estados Unidos foram usados para este estudo.

### 3.3.4 Transporte de carga terrestre

Projetos de rede do tipo E-R para problemas de transporte de carga são similares aos projetos para o transporte aéreo. Nesse problema o concentrador tem como objetivo servir de ponto de transferência e consolidação.

Segundo Camargo (2007) o projeto dessas redes, nesse contexto, podem variar de acordo com o tipo de transporte. Quando a carga é transportada utilizando toda a capacidade do veículo para um determinado ponto de demanda o problema é conhecido como *truckload*, ou seja, carga completa. Esse caso é mais adequado para transporte de demandas que requerem serviços personalizados e de grande volume. Já quando o veículo é carregado com cargas para ser entregue em diferentes pontos de demanda, o problema é conhecido como *less-than-truckload* ou carga fracionada. Nesse caso os clientes não necessitam ou não podem ser atendidos individualmente, dividindo assim o recurso de transporte com outros clientes, um exemplo são as transportadoras que entregam pequenos produtos comprados *on-line*.

No caso conhecido como *truckload* a preocupação maior no planejamento da rede não é com a consolidação da carga, mas na diminuição da duração da viagem do concentrador ao ponto de demanda. No outro caso, *less-than-truckload*, a preocupação é com a consolidação

da carga em volumes maiores para que a economia de escala seja obtida. Fatores como tempo máximo de duração de transporte, capacidade e tamanho da frota de veículos são elementos relevantes a serem considerados no modelo. Além disso, o problema aumenta sua complexidade devido a necessidade de rotear os veículos de forma eficiente para entrega das demandas.

Um estudo aplicado ao grupo SKF, uma empresa industrial localizada na Suécia, é feito por [Lumsden, Dallari e Ruggeri \(1998\)](#). Operando em cerca de 130 países, o Grupo SKF é líder mundial no negócio de rolamento, além de fornecer uma série de outros produtos industriais. Sua rede de vendas internacional é apoiada por cerca de 20.000 distribuidores e mais de 700 concessionários que atendem as principais áreas de clientes da SKF, como as indústrias automotiva, industrial, elétrica e mercados aeroespaciais. Uma análise feita pelos autores observou que a aplicação do conceito de redes do tipo E-R, a rede logística da empresa na distribuição europeia, permitiu uma economia próxima a 10% dos custos de transporte.

[Taylor et al. \(1999\)](#) fizeram um estudo de redes E-R aplicado a uma empresa de logística americana. Este estudo concentrou-se na melhoria de trabalho dos motoristas e nas necessidades de atendimento ao cliente. As estratégias apresentadas neste trabalho foram aplicadas dentro da rede de distribuição da empresa, diminuindo a rotatividade de motoristas, isto porque possibilitou melhores condições de trabalho.

[Cunha e Silva \(2007\)](#) apresentam um problema aplicado a uma empresa de transporte brasileira baseado em redes do tipo E-R com carga fracionada. A formulação proposta neste trabalho difere dos trabalhos encontradas na literatura no sentido de que sua função objetivo não-linear permite economias de escala para os custos de transporte que podem variar de acordo com a quantidade total de carga entre concentradores. Tal estratégia representa melhor o cenário em que este tipo de transporte está inserido.

[Chen, He e Fang \(2014\)](#) apresentam um estudo de caso, baseado em *less-than-truckload*, em uma empresa de logística chinesa. O trabalho teve como objetivo projetar e otimizar o funcionamento de uma rede baseada em E-R híbrida através da utilização de uma rede física E-R já existente. Os autores apontam que a rede híbrida foi mais vantajosa para o caso de estudo, sendo possível economizar 8% do custo total. Outro estudo baseado em um modelo híbrido pode ser encontrado no trabalho apresentado por [Zapfel e Wasner \(2002\)](#). A proposta foi aplicada a uma empresa austríaca de transporte, sendo constatado que foi possível diminuir 10% do custo total de transporte de longas distâncias. Os autores reforçam a ideia que a proposta poderia ser aplicada em outros contextos, e citam um problema encontrado na Alemanha.

## 3.4 Conclusão

O foco desta revisão bibliográfica foi mostrar como o problema de redes baseadas em E-R surgiu, onde tem sido aplicada, e quais são os principais métodos empregados para resolver o problema. As modelagens, bem como as formulações e métodos que têm sido utilizados para resolver os problemas abordados neste trabalho serão descritos nos próximos capítulos. Além disso, não enumeramos todos os trabalhos encontrados na literatura, foi utilizado somente aqueles que tem se destacado na literatura e influenciaram na elaboração deste trabalho.

## 4 Redes do tipo Eixo-Raio com alocação simples

Neste capítulo será detalhado o problema de redes do tipo E-R com alocação simples, custos fixos e não capacitado. Além disso, uma revisão bibliográfica listando os principais métodos para resolver o problema, bem como a heurística proposta são apresentados. Serão detalhados a representação computacional de uma solução, a função de avaliação, o método de construção, perturbações, os movimentos utilizados para a busca local e a estratégia adotada para o método *Path-Relinking*. Finalmente, os resultados e conclusões obtidas são apresentados.

### 4.1 Introdução

Esse problema pode ter diferentes definições. A que será abordada neste trabalho é baseada nas seguintes: Seja um conjunto  $N$  de nós de demanda e um conjunto  $K$  de nós candidatos a concentradores, onde  $n = |N|$ . Normalmente  $K \subseteq N$ , considerando que todos os nós de demanda são candidatos a concentradores, portanto  $K \equiv N$ . Para todo par de nós  $i$  e  $j$  ( $i, j \in N : i \neq j$ ),  $w_{ij}$  representa o fluxo de demanda entre o nó  $i$  (origem) e o nó  $j$  (destino) que é roteado por um ou dois concentradores. Normalmente  $w_{ij} \neq w_{ji}$ . Ainda, seja  $f_k$  o custo de instalação de um concentrador em um nó  $k \in N$  e  $c_{ijkm}$  o custo de transportar uma unidade de fluxo vindo da origem  $i$  com destino  $j$  sendo roteado pelos concentradores  $k$  e  $m$ , ou seja, o custo da rota  $i - k - m - j$ . O custo de transporte pode ser decomposto em três segmentos:  $c_{ijkm} = c_{ik} + \alpha c_{km} + c_{mj}$ , onde  $c_{ik}$  e  $c_{mj}$  são os custos de transporte do nó  $i$  até o concentrador  $k$  e do concentrador  $m$  até o nó  $j$ , respectivamente. Ainda temos  $\alpha c_{km}$  como o custo de transporte entre os concentradores  $k$  e  $m$ , nesse caso, ponderado por um fator  $\alpha$  que visa representar o fator de desconto, ou economia de escala, entre os nós concentradores. Se apenas um concentrador é utilizado, isso significa que  $k = m$ , nenhum fator de desconto é utilizado. Além disso, é considerado que existe uma aresta ligando todos os concentradores e o número de concentradores não é conhecido a priori. O objetivo é encontrar uma rede onde a soma do custo de instalação e distribuição de fluxo seja menor.

Um exemplo de uma rede do tipo Eixo-Raio é mostrado na Figura 22, onde os concentradores estão representados como triângulos, os pontos de demanda como círculos e as alocações são representadas pelas linhas, bem como as conexões entre os concentradores. Uma característica interessante dessa variante é a possibilidade de se explorar a relação entre custo fixo de instalação e custo de transporte, ou seja, quanto mais concentradores

instalados maior custo fixo de instalação e menor custo de transporte entre concentradores devido à economia de escala.



Figura 22 – Exemplo de uma rede do tipo E-R com alocação simples

Este problema pode ser formulado como:

$$\min \phi = \sum_k f_k z_{kk} + \sum_i \sum_{i \neq j} \sum_k \sum_m (w_{ij} c_{ijkm} + w_{ji} c_{jimk}) z_{ik} z_{jm} \quad (4.1)$$

s.a:

$$\sum_k z_{ik} = 1 \quad \forall i \in N \quad (4.2)$$

$$z_{ik} \leq z_{kk} \quad \forall i \neq j \in N \quad (4.3)$$

$$z_{ik} \in \{0, 1\} \quad \forall i, k \in N \quad (4.4)$$

, onde a variável  $z_{ik}$  indica que o nó  $i$  está alocado ao concentrador  $k$  caso seu valor seja 1, ou 0 caso contrário. Caso um nó seja concentrador, o valor de  $z_{kk} = 1$ . A função objetivo (4.1) busca minimizar os custos de transporte e instalação de concentradores. As restrições (4.2) asseguram que um nó será alocado a apenas um concentrador. As restrições (4.3) asseguram que os nós não concentradores serão atribuídos a somente nós concentradores. Finalmente, as restrições (4.4) consistem na integridade de domínio das variáveis. Existem diversas formulações para este problema, esta foi escolhida devida a sua simplicidade.

A primeira formulação matemática para esse problema foi proposta por O’Kelly (1992). Neste mesmo trabalho foram utilizadas metodologias para obter limites superiores e inferiores para o problema. Posteriormente, Campbell (1994c) propôs a primeira formulação linear inteira, uma vez que o trabalho proposto por O’Kelly (1992) consistia em uma formulação quadrática.

Abdinnour-Helm e Venkataramanan (1998) propuseram uma nova formulação inteira quadrática, baseada na ideia de fluxo de *multi commodities* em redes. Segundo os

autores, tal formulação se adequava melhor ao tipo de metodologia utilizada para resolver o problema, no caso o método *branch-and-bound* (B&B). Além disso, foi proposto um algoritmo genético capaz de encontrar soluções de qualidade mais rápido do que as metodologias da época.

Sohn e Park (1998) propuseram uma formulação inteira mista com melhorias. Os autores mostraram que a abordagem proposta poderia ser utilizada mesmo para grandes problemas, desde que o número de candidatos a concentradores fosse pequeno.

Um heurística híbrida, baseada em um algoritmo genético (AG) e busca tabu, foi proposta por Abdinnour-Helm (1998b). Enquanto o AG gerava soluções baseadas em alocação de concentradores a Busca Tabu procurava melhorar as alocações dos nós não concentradores. Foi usado um AG tradicional com representação binária para definir os concentradores. Já a Busca Tabu fez uso de dois tipos de movimento para efetuar a busca local. O primeiro, denominado *Exchange*, troca a alocação de dois nós em relação aos seus concentradores. Por exemplo, se o nó 2 é atribuído ao concentrador 1 e o nó 4 é atribuído ao concentrador 3, então as novas atribuições serão o nó 2 para o concentrador 3 e o nó 4 para o concentrador 1. Já o movimento *Shift* apenas muda a alocação de um nó não concentrador. A metodologia proposta foi superior ao AG aplicado ao mesmo problema sozinho, além disso, obteve os melhores resultados da literatura, até aquele ano.

Topcuoglu et al. (2005) propuseram uma heurística baseada em AG. Nessa proposta os indivíduos são gerados em três etapas: primeiro o número de concentradores são definidos aleatoriamente, dentro de um intervalo pré definido, para cada indivíduo; posteriormente, 70% do total de concentradores definidos no passo anterior são alocados de acordo com o fluxo de origem e destino dos candidatos a concentrador, os outros 30% são selecionados aleatoriamente. Por fim os nós não concentradores são atribuídos ao concentrador mais próximo. O operador de *crossover* usado é o tradicional para algoritmos genético. Dois operadores de mutação foram utilizados, *Shift* e *Exchange*, ambos foram considerados versões estendidas dos movimentos apresentados em Abdinnour-Helm (1998b). Essa proposta se mostrou mais eficiente do que o apresentado por Abdinnour-Helm (1998b), seu desempenho em relação a qualidade da solução e tempo de processamento foram consideravelmente superior.

Uma abordagem híbrida de AG e *Simulated Annealing* (SA) foi proposta por Cunha e Silva (2007). Nessa abordagem cada indivíduo do AG é submetido a um procedimento de busca local guiado pelo método SA. Assim como no trabalho apresentado por Abdinnour-Helm (1998b), o AG é responsável em gerar os concentradores, enquanto to SA faz uma busca local procurando a melhor alocação para um nó não concentrador. Os operadores utilizados para a busca local foram os mesmos usados em Topcuoglu et al. (2005). Resultados mostraram que a metodologia foi superior ao trabalho apresentado por Abdinnour-Helm (1998b).

Chen (2007) propôs uma heurística híbrida baseada nos métodos SA e Busca Tabu. O método de construção é dividido em três partes: determinação de número de concentradores, determinação de concentradores e alocação de não concentradores a concentradores. Inicialmente é utilizado um método baseado no custo de transporte e instalação para determinar um limite superior de concentradores para o problema. Para definir os concentradores iniciais são levados em consideração distância e fluxo entre os nós. Por fim, cada nó não concentrador é atribuído ao concentrador mais próximo. Para a busca local foram usados dois tipos de movimentos. O primeiro baseado no *Shift*, porém as trocas são efetuadas inicialmente com os nós que possuem menos fluxo. Segundo o autor, como cada nó não concentrador pode ser alocado a apenas um concentrador, há uma tendência de que o não concentrador com mais fluxo total ( $w_i$ ) de ser alocado para o concentrador mais próximo, enquanto a alocação do não concentrador com menos fluxo pode ser algo incerto. Portanto, o procedimento de melhoria é inicialmente aplicado a estes nós. O segundo movimento consiste em trocar o papel de dois nós, ou seja, um nó não concentrador torna-se concentrador, e o concentrador passa a ser um nó não concentrador. Movimentos para melhorar a solução obtida são executados baseado no método busca tabu, enquanto o critério de aceitação de uma solução e parada são baseados no SA. Os resultados obtidos mostraram que o método apresentou melhor desempenho em relação ao algoritmo apresentado por Topcuoglu et al. (2005).

Castro, Camargo e Miranda (2009) propuseram variantes do método de decomposição de Benders, aplicadas a instâncias com até 60 nós. Uma das variantes se mostrou competitiva quando comparada ao solver CPLEX, os autores salientaram que melhorias podem ser empregadas ao método para que seja possível resolver problemas ainda maiores. Tais melhorias foram empregadas em Castro (2010), assim o trabalho abordou instâncias com até 200 nós. Os resultados computacionais obtidos demonstram que os algoritmos desenvolvidos são competitivos quando comparados a qualidade da solução obtida e ao tamanho dos problemas resolvidos com tal metodologia. Além disso, o autor aponta que apenas as melhorias não é suficiente para um bom desempenho do método, é necessário que se tenha um modelo com uma boa relaxação linear.

Três variantes de métodos baseados nas metaheurísticas Multistart e Busca Tabu foram propostas por Silva e Cunha (2009). Basicamente os métodos a cada iteração geram uma solução que posteriormente é melhorada por uma busca local guiada pela meta-heurística busca tabu. O que difere os métodos é o modo de geração de uma solução inicial, que pode ser obtida das seguintes maneiras:

- para cada  $i \in N$  é sorteado um número aleatório entre 0 e 1, valores maiores que um parâmetro  $\rho$  de probabilidade fazem o nó  $i$  torna-se um concentrador.
- para cada  $i \in N$  é dada uma probabilidade  $\rho_i$  proporcional ao fluxo de entrada e

saída desse nó, ou seja, quanto maior o fluxo maior a probabilidade desse nó se tornar um concentrador.

- para cada  $i \in N$  é dada uma probabilidade  $\rho_i$  proporcional ao fluxo de entrada e saída desse nó, além disso, e considerado a distância entre o nó candidato e os outros nós da rede.

, após a definição dos concentradores, cada nó não concentrador é atribuído para o concentrador mais próximo. Uma busca local baseado no algoritmo Busca Tabu é executada após a criação da solução inicial. Neste trabalho apenas o movimento *Shift* foi usado na busca local. Como se tratada de uma heurística *Multi-Start*, a cada iteração são geradas novas soluções. Resultados mostraram que a metodologia proposta superaram os resultados da literatura até a data de publicação do artigo.

[Naeem e Berman \(2010\)](#) propuseram um AG com diferentes formas de codificação das soluções e operadores, quando comparado aos existentes na literatura. Os resultados mostraram que essa metodologia foi competitiva em comparação a outras encontradas na literatura.

Um método GRASP com reconexão por caminhos foi proposto em [Martins, Costa e Camargo \(2012\)](#). Para gerar a solução inicial foram utilizados os seguintes passos:

- passo 1: Escolha um nó  $i \in N$  aleatoriamente para se tornar concentrador, e calcule o custo da solução  $\zeta$ .
- passo 2: Para cada  $j \in N$  que não seja concentrador, torne-o um concentrador, realoque os nós não concentradores para o concentrador mais próximo e calcule o valor da solução. Caso o valor da solução seja menor que  $\zeta$ , adicione essa solução a uma lista de candidatos restrita (LRC)
- passo 3: Se a LCR estiver vazia, siga para o passo 4. Caso contrario, escolha aleatoriamente uma solução contida na LRC, atualize o valor de  $\zeta$  com o valor desta solução e volte ao passo 2.
- passo 4: Retorne a solução obtida.

Posteriormente, uma busca local com três diferentes estruturas de vizinhança é empregada a solução inicial obtida, nessa etapa a busca local foi guiada pelo método VND proposto por [Mladenovic e Hansen \(1997\)](#). As vizinhanças utilizadas foram *Shift*, "troca de concentrador", que visa trocar um concentrador por um nó não concentrador e realocar os não concentradores para os concentradores mais próximos e "remoção de concentrador", que visa retirar um nó concentrador da solução e posteriormente realocar os nós não concentradores aos concentradores mais próximos. Por fim, após algumas iterações do

método GRASP a técnica de reconexão por caminhos é aplicada utilizando um conjunto de soluções elites, esse conjunto contém as melhores soluções encontradas durante as iterações GRASP. Os resultados obtidos mostram que essa metodologia obteve os melhores resultados da literatura até aquela data.

Uma meta-heurística baseada em AG e busca local foi proposta por [Gomes et al. \(2013\)](#). Para gerar a população inicial para o AG foi utilizado uma adaptação do método proposto em [Martins, Costa e Camargo \(2012\)](#). Após a criação da população inicial, a solução que obteve o melhor valor de *fitness* é sujeita a uma busca local com o objetivo de melhorar a solução obtida. O restante do algoritmo segue os passos de um AG tradicional. Foram usados 5 tipos de operadores de mutação para o problema, baseados nos movimentos já conhecidos *Shift*, *Exchange*, troca de concentradores e remoção de concentradores. Além disso, foi usado um operador para adicionar um novo concentrador. Para a busca local é usado o movimento *Shift*. Os resultados obtidos mostraram que esse algoritmo superou todos os resultados de outros AG da literatura, até aquela data, para esse mesmo problema.

Outro AG com busca local foi proposto por [Maric, Stanimirovic e Stanojevic \(2013\)](#). Inicialmente, as soluções são criadas de forma pseudo-aleatório, a única preocupação nesse primeiro momento é garantir que a população seja heterogênea, em outras palavras, que cada indivíduo possua diferentes alocações de concentradores. Posteriormente, algumas soluções da população passam por uma busca local. Para o AG foram usados operadores tradicionais aplicados a codificações binárias. Além disso, táticas como elitismo e manutenção da diversidade foram aplicados. Já para a busca local, foram usados os movimentos *Shift* e "Troca de concentrador". Os resultados do método proposto foram melhores do que os reportados por [Silva e Cunha \(2009\)](#).

[Bailey, Ornbuki-Bernnan e Asobiela \(2013\)](#) propuseram um algoritmo baseado na meta-heurística Enxame de Partículas, do inglês *particle swarm optimization* (PSO). Até então nenhuma publicação tinha explorado essa meta-heurística para esse problema. Resultados mostraram que a metodologia é competitiva quando comparado com os melhores algoritmos da literatura em relação a qualidade de solução. O tempo computacional apenas foi comparado com uma estratégia usando um AG. Índícios mostraram que o tempo de convergência desse método é superior quando comparado a outros métodos.

[Ting e Wang \(2013\)](#) propuseram um heurística baseada no método *threshold accepting*, uma variação do método *Simulated Annealing* proposta por [Dueck e Scheuer \(1990\)](#). A solução inicial é criada a partir de um método que escolhe  $p$  concentradores baseado no fluxo de entrada, saída e custos de instalação dos nós. Foram utilizados cinco tipos de movimentos para geração de vizinhos. O primeiro troca a função de um concentrador com um nó não concentrador aleatoriamente. O segundo, adiciona um concentrador aleatoriamente, e realoca os nós para um concentrador mais próximo. O

terceiro, troca a função entre nós de um mesmo grupo, em outras palavras, entre um concentrador e os nós que estão alocados a ele. Posteriormente, os nós são alocados ao concentrador mais próximo. O quarto movimento consiste na troca de alocação dos nós não concentradores. Finalmente, o quinto movimento utiliza a ideia do primeiro movimento aplicado a  $p$  concentradores. Esses movimentos são utilizados no decorrer do algoritmo para gerar soluções vizinhas. Além disso, ao final é utilizado um método de busca local que utiliza os movimentos de "troca de função" e "realocação". Este algoritmo foi comparado a quatro outros trabalhos da literatura ([TOPCUOGLU et al., 2005](#); [CHEN, 2007](#); [CUNHA; SILVA, 2007](#); [SILVA; CUNHA, 2009](#)). Os resultados mostraram que o algoritmo se mostrou mais eficiente que estes algoritmos, tanto em tempo quanto qualidade das soluções.

[Brito, Gomes e Camargo \(2015\)](#) propuseram um método baseado nas meta-heurísticas GRASP, VND e VNS. Para gerar as soluções foi utilizado o método proposto em [Martins, Costa e Camargo \(2012\)](#), posteriormente para guiar o procedimento de busca local foram utilizados os métodos VND e VNS. Para o método VND foi utilizado uma vizinhança denominada "troca de concentrador". Para o método VNS foram exploradas as vizinhanças "remoção de concentrador", *Shift*, "insere concentrador" e "troca de concentrador". Os resultados obtidos mostraram que esse método foi superior ao trabalho proposto por [Gomes et al. \(2013\)](#). Além disso, essa metodologia foi submetida a um estudo de caso com o número de nós bem superior aos encontrados na literatura até a data da publicação do trabalho.

Um algoritmo baseado na meta-heurística Busca Tabu foi proposto em [Abyazi-Sani e Ghanbari \(2016\)](#). O método para geração de uma solução inicial considera todos os nós como concentradores e, a cada iteração, um nó concentrador é transformado em não concentrador, mais especificamente aquele que resulta na melhor função objetivo. Esse procedimento é repetido até que a retirada de um concentrador não resulte em melhora na função objetivo. Para busca local, foram utilizados os movimentos "inserir um concentrador", "retirar concentrador" e *Shift*. Resultados mostraram que o método foi muito eficiente, obtendo resultados superiores quando comparado a outras metodologias.

Embora existam vários pontos positivos nos métodos apresentados anteriormente, tais como a eficiência do método proposto por [Abyazi-Sani e Ghanbari \(2016\)](#), a construção proposta por [Martins, Costa e Camargo \(2012\)](#), ou a capacidade de exploração do algoritmo Memético proposto por [Gomes et al. \(2013\)](#), quando aplicados a problemas de grande escala eles podem ter seus desempenhos comprometidos. Isto porque requerem uma quantidade de processamento (ou tempo computacional) significativamente grande para obter uma solução de qualidade. Por exemplo, o método de construção proposto por [Martins, Costa e Camargo \(2012\)](#) gasta menos de 1 segundo para gerar uma solução para uma instância de 100 nós, já para um problema com 3000 nós pode demandar mais de 2000 segundos. Assim, a proposta de métodos robustos que possam tratar problemas em diferentes escalas,

dificuldade ou contexto, de forma eficiente, ainda é um grande desafio.

O restante deste capítulo mostra que a programação paralela pode ajudar na concepção de uma heurística escalável, ou seja, capaz de lidar com problemas em diferentes escalas, sem perder eficiência no que se diz respeito a qualidade da solução e tempo para alcançá-la. Como visto anteriormente, as atuais metodologias aplicadas a este problema não exploraram este recurso, o que podem limitar suas aplicações. Geralmente, para problemas de grande escala restritos a pequenos limites de tempo para processamento não há um bom comportamento destas metodologias. Não menos importante, o uso da paralelização pode permitir alcançar soluções que não são facilmente alcançáveis por métodos sequenciais, isso devido ao sistema colaborativo e a capacidade de explorar um número maior de soluções do espaço de busca em um mesmo intervalo de tempo. A seguir serão especificados os detalhes de implementação das variantes do ILS-RVND-PR, descrito no Capítulo 2, e apresentados os resultados e conclusões obtidos.

## 4.2 Metodologia

Nesta seção será detalhada a heurística proposta para resolver o problema. Vale ressaltar que apenas serão detalhados as partes que compõem a heurística baseada no método ILS-RVND-PR, descrito no capítulo 2.

### 4.2.1 Representação Computacional

Para representar uma solução são utilizados dois vetores  $S^C$  e  $S^A$ , ambos com tamanho  $n$ , representando, respectivamente, os concentradores instalados e as alocações.  $S_i^C$  é binário, e a  $i$ -ésima posição do vetor indica se o nó  $i \in N$  é concentrador. Neste caso, as posições contendo nós concentradores são assinaladas com o valor 1, e as posições com nós não concentradores são assinaladas com 0. Já o vetor  $S_i^A$  é inteiro e indica a qual concentrador o nó  $i \in N$  está alocado. A Figura 23 ilustra um exemplo desta representação.

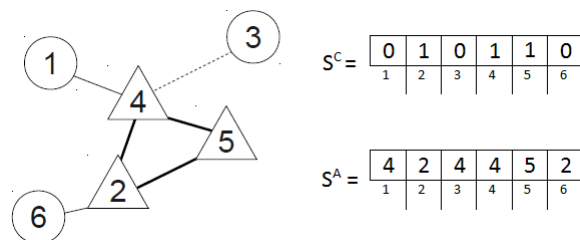


Figura 23 – Representação computacional para o problema de localização de concentradores com alocação simples.

Os nós concentradores são representados pelos triângulos, é possível perceber que no vetor  $S^C$  apenas os nós 2, 4 e 5 estão assinalados com o valor 1. Já as alocações são

representadas pelo vetor  $S^A$ , neste caso cada nó concentrador está alocado a ele mesmo, e os demais estão alocados de acordo com a Figura 23.

## 4.2.2 Método de construção

A ideia geral dos métodos de construção implementados neste trabalho é baseada nos algoritmos de agrupamento hierárquico. Nestes algoritmos os grupos são formados através de aglomeração ou divisão. No primeiro caso, aglomeração, inicialmente cada elemento pertence a um grupo distinto, que posteriormente são agrupados, considerando alguma métrica de similaridade, até que um critério de parada seja satisfeito. Já na abordagem baseada em divisão, inicialmente todos elementos pertencem a um mesmo grupo, que gradativamente vão se dividindo em grupos diferentes, baseado em uma métrica de similaridade, até que um critério de parada seja satisfeito.

### 4.2.2.1 Método 1 (M1)

O primeiro método consiste em escolher um concentrador inicial para compor uma solução  $S$  e, posteriormente, melhora-la iterativamente, inserindo concentradores adicionais até que o custo da solução não possa ser melhorado. Considerando dois conjuntos  $O_s$  e  $C_s$  representando, respectivamente, o conjunto dos nós concentradores e não concentradores de uma solução  $S$ . Ainda, sendo  $O_s \cap C_s = \emptyset$  e  $O_s \cup C_s = N$ . O primeiro nó  $j \in C_s$  escolhido para se tornar concentrador é aquele que retorna o menor valor de função objetivo  $\phi$ .

Posteriormente, para cada nó  $j \in C_s$ , é calculada a variação no valor da função objetivo ( $\delta_j$ ) após transformar  $j$  em concentrador e realocar todos os nós não concentradores aos concentradores mais próximos. Quando  $\delta_j \leq 0$ , o nó  $j$  passa a ser um candidato para fazer parte do conjunto  $O_s$ . A Figura 24 mostra os primeiros passos para construir uma solução. Considere que inicialmente o nó 3 foi escolhido como concentrador (Figura. 24(a)). Posteriormente, como ilustrado na Figura 24(b), são calculadas as variações de função objetivo ao inserir novos concentradores a solução. É possível observar duas situações, a primeira onde  $\delta_1$  tem valor positivo, isso significa que ao tornar o nó 1 concentrador ele incrementa em uma unidade a função objetivo. Para os outros nós, onde  $\delta_j$  é negativo, a transformação de um nó  $j$  em concentrador diminui o valor da função objetivo de acordo com o valor indicado. Por exemplo, ao tornar o nó 5 um concentrador o valor da função objetivo reduz 8 unidades.

O cálculo de  $\delta_j$  é feito de acordo com a equação (4.6). Para todo nó  $i \in N$  que for alocado ao novo concentrador  $j$ , baseado no critério de distância, ou seja, se a distância do atual concentrador do nó  $i$  ( $c_{iH_i}$ ) for maior em relação ao novo concentrador  $j$  ( $c_{ij}$ ), é calculado a variação na função objetivo através de  $\Delta_i$ . Considere  $H = S^A$ , assim o concentrador onde o nó  $i$  estava alocado pode ser representado por  $H_i$ , e  $Q$  o conjunto de todos os nós que foram realocados. O objetivo é recalculer somente parte da função

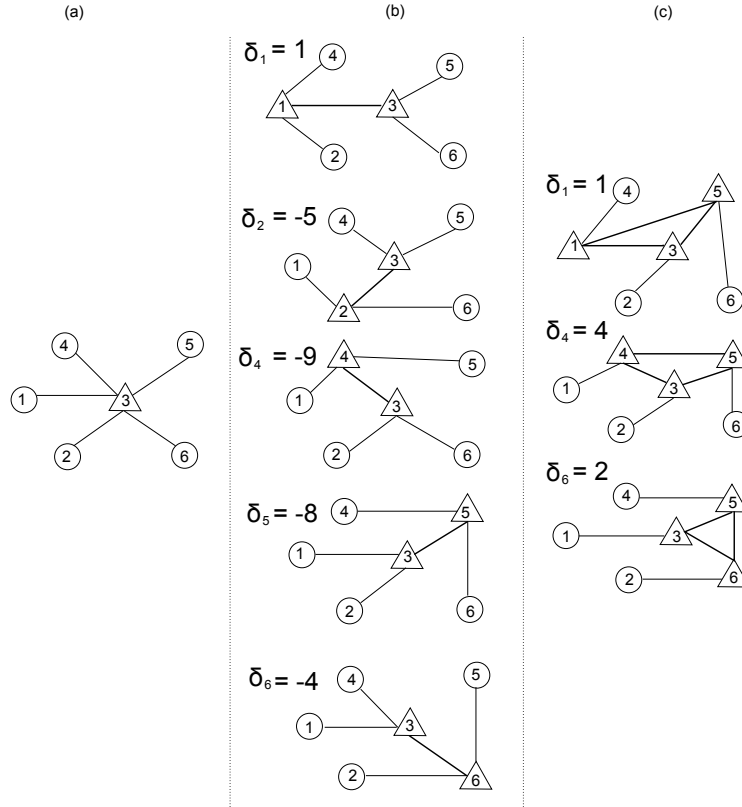


Figura 24 – Exemplo de uma iteração do método de construção

objetivo que sofre alteração.

$$\Delta_i = \sum_l^n w_{il}(c_{ij} - c_{iH_i}) + w_{li}(c_{ji} - c_{H_i i}) + \alpha \left[ w_{il}(c_{jH_i} - c_{H_i H_i}) + w_{li}(c_{H_i j} - c_{H_i H_i}) \right] \quad (4.5)$$

$$\delta_j = f_j + \sum_{i \in Q} \Delta_i \quad (4.6)$$

Após o cálculo de cada  $\delta_j (j \in C_s)$ , é escolhido aquele que retorna o menor valor, de acordo com a equação (4.7). Caso exista um nó  $j$  que atenda esta condição, o seguinte procedimento é executado:  $O_s = O_s \cup \{j\}$  e  $C_s = C_s \setminus \{j\}$ .

$$j = \arg \min_{j \in O_s} \{\delta_j : \delta_j < 0\} \quad (4.7)$$

Nas iterações seguintes são recalculados  $\delta_j$ , para todo  $j \in C_s$ , e escolhido aquele que mais contribui na melhora da função objetivo. O procedimento continua até que não seja possível melhorar a função objetivo, ou seja,  $\delta_j \geq 0$  para todo  $j \in C_s$  (ver algoritmo 1). Esse cenário é ilustrado na Figura 24(c).

Este procedimento (ver Algoritmo 2) é baseado no método proposto por [Gomes et al. \(2013\)](#), sendo diferente o modo de escolha do primeiro nó para compor a solução e por se tratar de um algoritmo guloso, diferentemente da outra abordagem que é baseado no método de construção GRASP. Os cálculos dos  $\delta_j (j \in C_s)$  são independentes, isso permite

que sejam processados paralelamente para cada  $j$ . Assim sendo, o algoritmo proposto faz uso de todos os núcleos disponíveis para acelerar a fase de construção, o que não é feito na abordagem anterior.

---

**Algoritmo 2: MÉTODO 1**

---

```

1 início
2    $j \leftarrow \arg \min_{j \in C_s} \{\phi_j\}$ 
3    $O_s \leftarrow \{j\}$ 
4    $C_s \leftarrow C_s \setminus \{j\}$ 
5    $f \leftarrow \phi_j$ 
6   repita
7      $j \leftarrow \arg \min_{j \in C_s} \{\delta_j\}$ 
8     Se  $\delta_j < 0$  então
9        $O_s \leftarrow O_s \cup \{j\}$ 
10       $C_s \leftarrow C_s \setminus \{j\}$ 
11       $f \leftarrow f + \delta_j$ 
12     fim
13   até  $\delta_j \geq 0$ ;
14 fim

```

---

#### 4.2.2.2 Método 2 (M2)

O segundo método consiste em definir cada nó como um concentrador para compor a solução inicial  $S$ , e posteriormente, melhora-la, iterativamente, removendo os concentradores, até que o custo da solução não possa ser melhorado. Assim sendo, inicialmente  $O_s = N$  e  $C_s = \emptyset$ . A cada iteração, para cada  $j \in O_s$ , é calculado a variação no valor da função objetivo  $\delta'_j$  de acordo com a equação (4.8), após retirar o concentrador  $j$  da solução, e realocar todos os nós que estavam alocados a  $j$  (incluindo  $j$ ) ao concentrador mais próximo. Para a realocação é utilizada uma matriz auxiliar  $c_{ij}^{aux}$ , contendo em ordem crescente a informação dos nós que estão mais próximos de um determinado nó  $i$ . Por exemplo, sendo  $i = 1$ , a posição  $c_{11}^{aux}$  teria a informação do nó mais próximo de  $i$ , a posição  $c_{12}^{aux}$  a informação do segundo nó mais próximo, e assim sucessivamente. Assim a realocação é feita para o nó mais próximo que é concentrador.

$$\delta'_j = -f_j + \sum_{i \in Q} \Delta_i \quad (4.8)$$

Novamente, assim como no Método 1, é escolhido o nó  $j$  de acordo com a equação (4.7). Assim, caso exista um nó  $j$  que atenda estas condições, o seguinte procedimento deve ser executado:  $O_s = O_s \setminus \{j\}$  e  $C_s = C_s \cup \{j\}$ . A Figura 25 (a) ilustra o cenário inicial

do método, onde todos os nós são concentradores. A Figura 25(b) ilustra o cenário (ou segunda iteração) onde é verificado o impacto ao se retirar um nó  $j$  do conjunto  $O_s$ . Neste caso, os valores de  $j = \{1, 3\}$  fazem com que a função objetivo tenha a mesma redução, em três unidades. O procedimento continua até que não exista  $\delta'_j < 0$ , o que é ilustrado no Figura 25 (c).

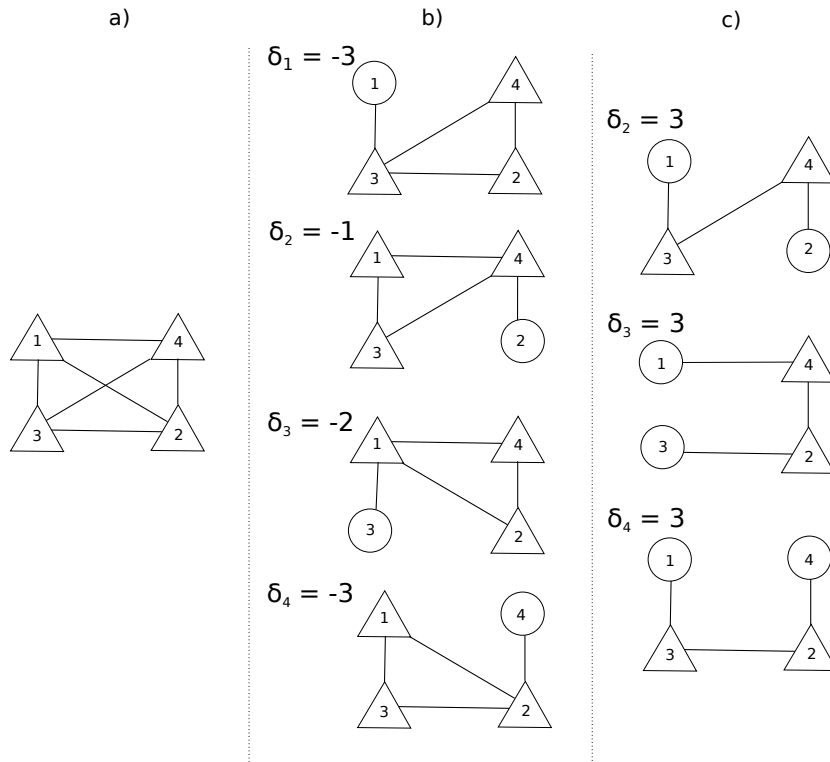


Figura 25 – Exemplo de uma iteração do método de construção

Este procedimento (ver algoritmo 3) é baseado no método proposto por [Abyazi-Sani e Ghanbari \(2016\)](#), sendo diferente neste trabalho os cálculos de  $\delta'$ , que são implementados de forma paralela.

### 4.2.3 Estruturas de vizinhança

Existe uma grande variedade de estruturas de vizinhança que podem ser exploradas, pelos métodos de busca local, para o problema tratado neste trabalho. Neste trabalho foram utilizados quatro estruturas, denominadas: “Realocação”, ”Troca de Função”, “Remover Concentrador” e “Adicionar Concentrador”. Estas estruturas são descritas a seguir:

- “Realocação”: para cada nó  $j \in C_s$  é feita a tentativa de realoca-lo a um concentrador  $k \in O_s$ . O número de vizinhos gerados a partir dessa vizinhança é  $|C_s|(|O_s| - 1)$ , e a quantidade de operações (no pior caso) para avaliar todas as soluções é aproximadamente  $2n(|C_s|(|O_s| - 1))$ ;

**Algoritmo 3: MÉTODO 2**


---

```

1 início
2    $O_s \leftarrow \{N\}$ 
3    $C_s \leftarrow \{\}$ 
4    $f \leftarrow \phi_j$ 
5   repita
6      $j \leftarrow \arg \min_{j \in C_s} \{\delta'_j\}$ 
7     Se  $\delta_j < 0$  então
8        $C_s \leftarrow C_s \cup \{j\}$ 
9        $O_s \leftarrow O_s \setminus \{j\}$ 
10       $f \leftarrow f + \delta'_j$ 
11     fim
12   até  $\delta'_j \geq 0$ ;
13 fim

```

---

- “Troca de Função”: cada concentrador  $k \in O_s$  é selecionado, um a um, para se tornar um nó não concentrador, e um nó não concentrador alocado a  $k$  é escolhido para se tornar um concentrador. Em seguida, os nós não concentradores que estavam alocados a  $k$  são realocados para o concentrador que reduz os custos de transporte. O número de vizinhos gerados a partir dessa vizinhança é  $|C_s|$ , e a quantidade de operações (no pior caso) para avaliar todas as soluções é aproximadamente  $2n^2|C_s|$ ;
- “Remover Concentrador”: os concentradores  $k \in O_s$  são removidos um a um, ou seja,  $O_s = O_s \setminus k$  e  $C_s = C_s \cup k$ . Para cada remoção, todos os nós alocados a  $k$  são atribuídos para o concentrador que reduz os custos de transporte. O número de vizinhos gerados a partir dessa vizinhança é  $|O_s|$ , e a quantidade de operações (no pior caso) para avaliar todas as soluções é aproximadamente  $2n^2(|O_s| - 1)$ ;
- “Adicionar Concentrador”, cada nó  $j \in C_s$  é escolhido para se tornar concentrador, então é verificado se a alocação dos nós não concentradores a este novo concentrador melhora a função objetivo. O número de vizinhos gerados a partir dessa vizinhança é  $|C_s|$ , e a quantidade de operações (no pior caso) para avaliar todas as soluções é aproximadamente  $2n|C_s|^2 - 2n|C_s|$ .

O critério utilizado para explorar estas estruturas de vizinhança com o método ILS-RVND1 é conhecido como Primeiro Aprimorante, exceto para a estrutura de vizinhança “Realocação”, no qual o Melhor Aprimorante é utilizado. Esta estratégia foi adotada porque obteve o melhor desempenho em testes preliminares. Além disso, utilizar o critério melhor Aprimorante em alguns tipos de vizinhança pode ser computacionalmente caro. A versão ILS-RVND2, utiliza para todas as vizinhanças o critério Melhor Aprimorante. O custo em utilizar tal critério é compensado ao se explorar paralelamente estas estruturas.

#### 4.2.4 Perturbação

Conforme explicado no Capítulo 2, o procedimento de perturbação utilizado neste trabalho restringe a cada *thread* de busca algumas posições do vetor solução para aplicar a perturbação. Os movimentos utilizados para perturbar uma solução são detalhados a seguir:

- Troça de Função: Este movimento pode alterar a estrutura de uma solução de duas formas: a) transformando um nó não concentrador em um nó concentrador; b) transformando um nó concentrador em um nó não concentrador. Em ambos os casos, todos os nós não concentradores são alocados aos concentradores mais próximos.
- Realocação: Este movimento altera a estrutura de uma solução mudando a alocação dos nós não concentradores aleatoriamente.

Além disso, a variável  $\pi$  define a probabilidade de cada posição de uma solução sofrer alteração. Quanto maior este valor, maior a probabilidade de um dos movimentos ser executado. Os movimentos são escolhidos aleatoriamente.

#### 4.2.5 Path-Relinking

Neste trabalho, o método Path-Relinking (PR) é aplicado a duas soluções, no qual denominaremos de solução guia e solução guiada. A cada iteração, são inseridos ou removidos concentradores que estão na solução guia, mas que não estão na solução guiada. A cada inserção/remoção uma busca local utilizando a estrutura de vizinhança “Realocação” é utilizada. No fim de cada iteração, o melhor movimento é implementado, mesmo que não melhore a solução guiada. Sempre que o processo PR encontra uma solução melhor do que a melhor solução global ( $s^*$ ), esta solução é atualizada. O método para somente quando as soluções guia e guiada têm a mesma estrutura de concentradores.

A Figura 26 exemplifica a ideia do método PR. Considere a Figura 26 (a) como a solução guiada, e a Figura 26 (d) como a solução guia. Podem ser geradas diferentes soluções inserindo ou removendo concentradores com o objetivo de transformar a solução guiada na solução guia. Por exemplo, a Figura 26 (b), ou na primeira iteração do método PR, a primeira estrutura é criada estabelecendo o nó 1 como um concentrador, enquanto a segunda e a terceira são obtidas removendo os concentradores 2 e 5, respectivamente. Assumindo que o melhor movimento desta iteração é o que gera a terceira estrutura da Figura 26 (b), a segunda iteração começa então a partir desta solução. São agora possíveis dois movimentos diferentes: A inserção do concentrador 1 e a remoção do concentrador 2, a primeira e segunda estrutura ilustradas na Figura 26(c).

Novamente, assumindo que a primeira estrutura é o resultado do melhor movimento nesta iteração, existe apenas um movimento que transforma a solução guiada na solução

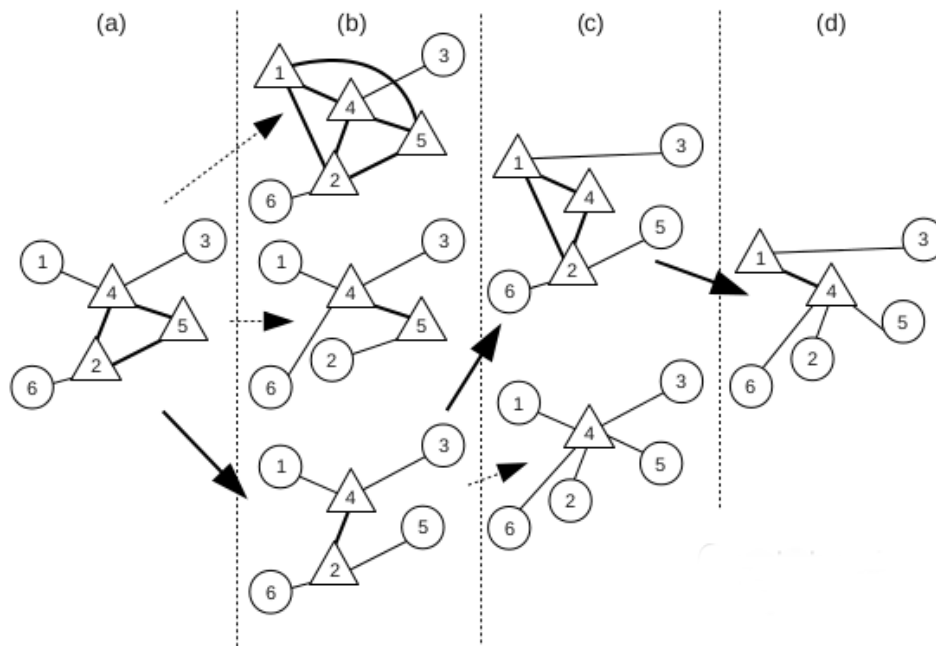


Figura 26 – Ilustração do método *Path-Relinking*

guia, neste caso a remoção do concentrador 2. O método para quando o conjunto  $O_s$  das soluções guiada e guia são iguais.

Atualmente existem diversas formas para exploração das soluções geradas quando aplicado o método PR. Estas variam em relação ao número de soluções exploradas, definição das soluções guia e guiada e escolha do caminho para exploração. A que será utilizada neste trabalho é conhecida como *backward and forward relinking*, pois apresentou os melhores resultados em teste preliminares.

### 4.3 Experimentos Computacionais

Esta seção apresenta os resultados obtidos pelos métodos propostos neste trabalho. Além disso, é feita uma comparação de desempenho destes algoritmos em relação a melhor heurísticas conhecida da literatura. Foram utilizados três computadores nos experimentos. O primeiro, que será chamado de *PC1*, consiste em um computador com um único processador de dois núcleos (4 *threads*), modelo Intel i5-3537U de 1.8Ghz com 4Gb de memória RAM. O segundo, que será chamado de *PC2*, consiste em um computador com um único processador de 4 núcleos (8 *threads*), modelo Intel i7-4790 de 3.6 Ghz com 15Gb RAM. Já o terceiro, que será chamado de *PC3*, consiste em um computador com dois processadores, cada processador contando com 6 núcleos (24 *threads*), modelo Intel Xeon E5-2620 de 2.1 GHz com 64 Gb RAM. Ambas as máquinas executando o sistema operacional baseado em Unix (Linux).

### 4.3.1 Método de construção

Nesta seção, são apresentados os resultados obtidos pelos métodos de construção sequenciais e paralelos. Todos os experimentos desta seção foram feitos com o uso do computador *PC3*. A implementação dos cálculos em paralelo no procedimento de construção permitiu um ganho considerável em relação ao tempo a partir de instâncias onde  $n > 100$ . A Figura 27 ilustra o comportamento do *speedup* para os dois métodos de construção em relação ao tamanho das instâncias. Considere o eixo das abscissas sendo o número de nós das instâncias tratadas neste trabalho, enquanto o eixo das ordenadas o valor do *speedup*. A linha tracejada e contínua representam, respectivamente, os Métodos 1 e 2. Um algoritmo eficiente é aquele que obtém valores altos para o indicador *speedup*. De maneira geral, é possível perceber que para pequenos valores de  $n$  os métodos paralelos são menos eficientes que os sequenciais, pois valores menores que 1 mostram que o algoritmo sequencial é mais rápido. Isso pode ser explicado devido ao fato de que para ocorrer ganho de desempenho, o *speedup* obtido pela região paralelizada deve se sobrepôr ao *overhead* de inicialização e gerenciamento das *threads*.

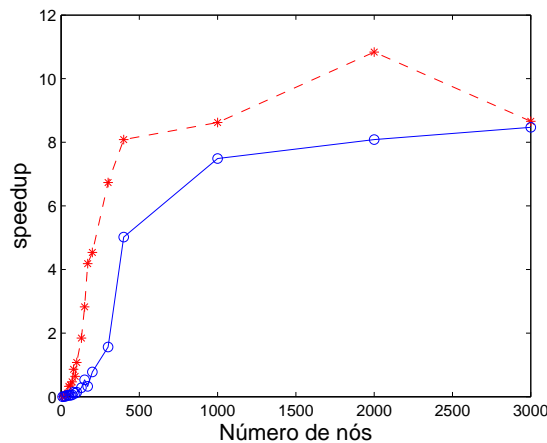


Figura 27 – *speedup* calculado para o método de construção paralelo em relação ao sequencial

Estes métodos de construção contrariam a ideia de que se for usado o dobro de recursos de *hardware*, no caso núcleos ou *threads*, eles seriam duas vezes mais rápidos. Na verdade isto raramente acontece, devido ao fato de existir uma variedade de custos associados ao paralelismo. Além de executar a computação essencial (isto é, a computação que seria executada pelo programa serial para resolver a mesma instância de problema), um programa paralelo também pode gastar tempo na comunicação entre processos/*threads*, sincronização de tarefas, na criação das próprias *threads*, esperas ociosas em regiões críticas, entre outros. Assim, pode-se dizer que o tempo dispensado para coordenar tarefas que não seja a computação essencial é chamado de *overhead*.

Além disso, segundo a lei de Amdahl, o ganho de desempenho de um algoritmo

paralelo (determinístico) é limitado pela parte sequencial do mesmo, e pode ser calculada a partir da equação 4.9 (STALLINGS, 2010). Considere  $f$  sendo a parte que pode ser "infinitamente" paralelizável, sem nenhum *overhead*,  $(1 - f)$  a parte sequencial do mesmo e  $q$  o número de núcleos disponíveis. Assim, em um sistema *multicore* com 8 núcleos e apenas 10% de código sequencial, pode produzir *speedup* máximo de até 4,7. Aumentando a parte sequencial para 20%, o *speedup* diminui para 3,3. Vale ressaltar que estes são valores máximos, sem considerar qualquer tipo de *overhead*.

$$\text{speedup máximo} = \frac{1}{(1 - f) + \frac{f}{q}} \quad (4.9)$$

Ainda, considerando o desempenho dos métodos de construção, a Figura 27 mostra que há um perda de desempenho, mais especificamente de *speedup*, entre as instâncias onde  $2000 \leq n \leq 3038$ , para o método 1. Segundo Stallings (2010), algoritmos paralelos podem gerar *overhead* significativo como resultado de comunicação e distribuição de trabalho para vários processadores, e para coerência de cache. Isso é refletido em uma curva de desempenho (*speedup*) na qual se alcança picos, e posteriormente, a perda de desempenho por *overhead*. Na literatura é possível encontrar trabalhos com o objetivo de explicar tal comportamento, entre os motivos principais, se destaca o uso da memória cache dos processadores. De acordo com Grama et al. (2003), este tipo de memória influencia diretamente no comportamento de algoritmo paralelos, uma vez que os múltiplos núcleos podem compartilhar este recurso, e por isso, o seu gerenciamento pode ser custoso. Além disso, devido o tamanho dos dados, mais acessos a memória primária precisam ser feitos, o que é mais custoso do que o acesso a memória cache, causando *overhead*. Mais informações sobre a relação memória cache e o desempenho de algoritmos paralelos podem ser obtidos nos trabalhos apresentados por: Al-Shandawely (2010); Pais (2014) e Pereira (2015).

Assim, algumas características das instâncias com 3038 nós podem ter levado a perda de desempenho, tais como: tamanho da instância (quantidade de memória utilizada) e quantidade de processamento necessário para obter a solução. Estas características vão de encontro com o que foi explanado anteriormente, tanto em relação a memória quanto em relação a comunicação e distribuição de trabalho (processamento) entre os processadores.

Comparando os algoritmos, algumas características importante devem ser observadas. O algoritmo  $M1$ , tanto a versão paralela, quanto a versão sequencial, é mais rápido que o algoritmo  $M2$ , para a maioria das instâncias (ver tabela 2). Devido a semelhança na implementação dos métodos de construção, foi analisado a relação entre o tempo para se obter uma solução e a quantidade de avaliações de função. Verificamos que no pior caso, a quantidade de avaliações de função de  $M2$  chega a ser 2,3 vezes maior. Nas instâncias de 3038 nós, o comportamento dos métodos são diferentes,  $M1$  chega a fazer 18% a mais de avaliações que  $M2$ , o que pode explicar o tempo superior para obter uma solução.

Tabela 1 – Número de avaliações de função para se obter uma solução para alguns problemas tratados neste trabalho.

Instância	$M1$	$M2$	Instância	$M1$	$M2$
ap20_2	220	513	BR2_200_2	39090	57521
ap40_4	1053	2070	BR3_300_2	94905	131271
ap50_6	5758	8621	BR4_300_6	67564	131271
ap90_8	4919	11748	BR5_402_8	141215	237402
ap100_2	11202	13771	BR6_402_2	198316	235809
ap100_4	5535	10873	BR7_1000_4	1511159	1497348
ap100_6	6931	13771	BR8_2000_2	6676218	5915231
ap100_8	5970	14359	BR9_3038_2	16660005	13618538

A tabela 1 mostra o número de avaliações de função que são necessárias, para cada método, para a construção de solução considerando algumas instâncias utilizadas neste trabalho. Considerando ainda as tabelas 1 e 2, é possível observar que o número de avaliações de função tem impacto direto no tempo para se obter uma solução, e que um número pequeno deste procedimento faz com que o algoritmo sequencial seja mais rápido.

Além disso, vale observar que embora o método  $M1$  faça menos iterações, cada iteração tem um custo maior relacionado a análise de funções. A Figura 28 ilustra este comportamento, considere o limite do eixo das abscissas sendo o número de iterações do algoritmo. Além disso, a linha pontilhada representando  $M1$ , enquanto a linha contínua  $M2$ . Como pode ser observado, a diferença da quantidade de análises de função, entre  $M1$  e  $M2$ , diminui a medida que o valor de  $n$  aumenta, até que  $M2$  passa a ser mais eficiente.

Quando a questão é qualidade das soluções não existe diferença entre as versões sequencial e paralela de cada método, ou seja  $M1^p \times M1^s$  e  $M2^p \times M2^s$ . Isso porque foi utilizada a ideia de paralelização a nível de iteração. Se tratando de  $M1 \times M2$ , foi observado que  $M2$  obtêm soluções de melhor qualidade, na maioria dos casos, sendo até 1% superior. Não menos importante, para instâncias onde o ótimo é conhecido, as soluções possuem um *gap* de aproximadamente 2%. A tabela 3 mostra o valor de função objetivo obtidos pelos métodos  $M1$  e  $M2$  para cada instância tratada neste trabalho.

Considerando todas as características descritas anteriormente, será utilizado nos demais experimentos a construção  $M2$ . Embora ela seja, na maioria dos casos, mais lenta que  $M1$ , o custo para obter uma solução de melhor qualidade é válido, pois melhora a convergência do método de busca para algumas instâncias. Além disso, os resultados apontam que este método é mais escalável, em outras palavras, para instâncias maiores que as tratadas neste trabalho existem alguns indícios que mostram que ela teria um melhor comportamento.

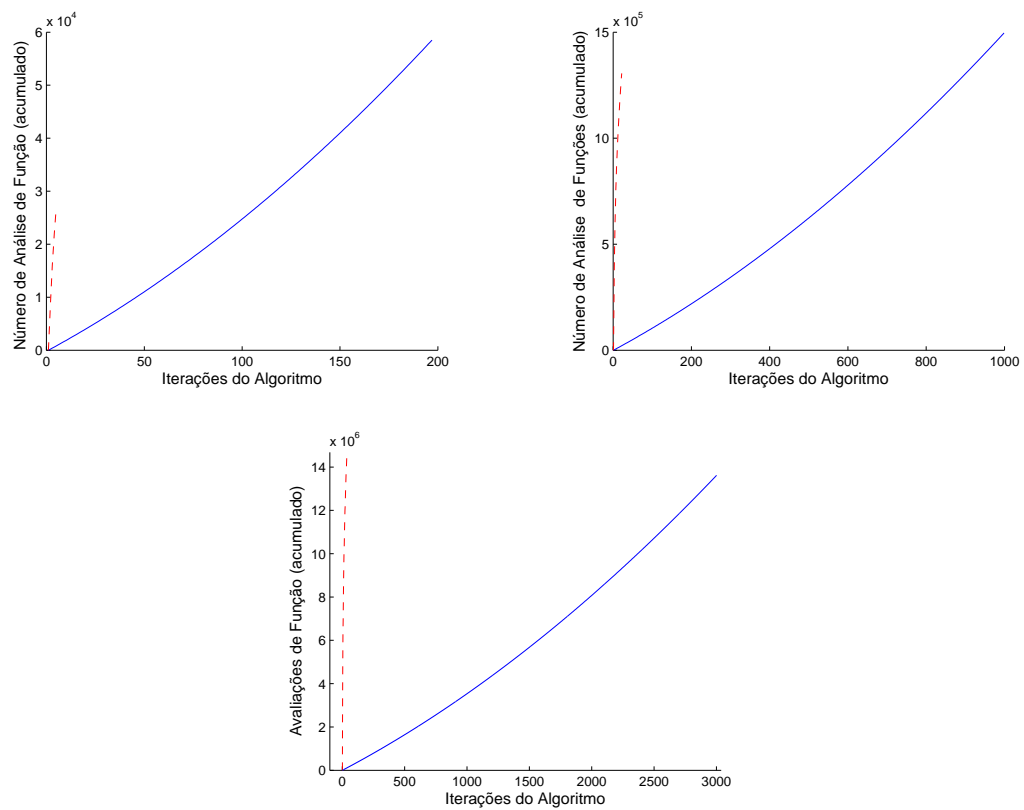


Figura 28 – Quantidade de avaliações de função e iterações para obter a solução inicial.

Tabela 2 – Tempo médio em segundos para criação de uma solução utilizando os métodos de construção paralelo e sequencial.

Instância	$M1^P$	$M2^P$	$M1^s$	$M2^s$	Instância	$M1^P$	$M2^P$	$M1^s$	$M2^s$	Instância	$M1^P$	$M2^P$	$M1^s$	$M2^s$
ap10_2	0,027	0,004	0	0	ap80_2	0,041	0,374	0,024	0,029	BR1_100_0.2	0,036	0,334	0,04	0,058
ap10_4	0,016	0,046	0	0	ap80_4	0,034	0,015	0,025	0,03	BR1_100_0.4	0,032	0,483	0,03	0,048
ap10_6	0,009	0,045	0	0	ap80_6	0,005	0,372	0,015	0,025	BR1_100_0.6	0,03	0,481	0,032	0,059
ap10_8	0,012	0,044	0	0	ap80_8	0,008	0,013	0,012	0,029	BR1_100_0.8	0,026	0,485	0,036	0,06
ap20_2	0,006	0,092	0	0	ap90_2	0,051	0,09	0,034	0,043	BR2_200_0.2	0,062	0,879	0,319	0,432
ap20_4	0,016	0,005	0	0	ap90_4	0,047	0,417	0,028	0,035	BR2_200_0.4	0,055	1,004	0,281	0,421
ap20_6	0,012	0,097	0	0,001	ap90_6	0,034	0,423	0,023	0,035	BR2_200_0.6	0,046	0,61	0,276	0,428
ap20_8	0,013	0,094	0	0,001	ap90_8	0,027	0,442	0,017	0,036	BR2_200_0.8	0,048	0,064	0,249	0,449
ap30_2	0,035	0,127	0,001	0,002	ap100_2	0,051	0,47	0,035	0,057	BR3_300_0.2	0,148	0,536	0,924	1,383
ap30_4	0,006	0,142	0,001	0,003	ap100_4	0,048	0,473	0,034	0,046	BR3_300_0.4	0,158	1,083	0,983	1,363
ap30_6	0,007	0,145	0,001	0,001	ap100_6	0,009	0,375	0,039	0,058	BR3_300_0.6	0,153	1,542	1,083	1,361
ap30_8	0,019	0,141	0,001	0,003	ap100_8	0,024	0,014	0,029	0,048	BR3_300_0.8	0,145	0,669	1,004	1,362
ap40_2	0,039	0,001	0,002	0,006	ap130_2	0,078	0,199	0,111	0,125	BR4_300_0.2	0,162	0,176	1,053	1,35
ap40_4	0,032	0,177	0,004	0,003	ap130_4	0,016	0,165	0,108	0,125	BR4_300_0.4	0,156	0,641	0,997	1,394
ap40_6	0,006	0,17	0,002	0,005	ap130_6	0,069	0,099	0,085	0,103	BR4_300_0.6	0,134	1,094	1,045	1,411
ap40_8	0,021	0,055	0,003	0,005	ap130_8	0,04	0,156	0,07	0,104	BR4_300_0.8	0,115	0,616	0,798	1,352
ap50_2	0,037	0,204	0,008	0,011	ap150_2	0,085	0,704	0,174	0,187	BR5_402_0.2	0,308	0,708	2,623	3,389
ap50_4	0,017	0,201	0,008	0,011	ap150_4	0,027	0,028	0,146	0,153	BR5_402_0.4	0,313	0,583	2,578	3,37
ap50_6	0,029	0,189	0,007	0,006	ap150_6	0,09	0,399	0,177	0,195	BR5_402_0.6	0,319	0,859	2,631	3,391
ap50_8	0,006	0,198	0,006	0,007	ap150_8	0,022	0,248	0,136	0,195	BR5_402_0.8	0,325	0,956	2,438	3,389
ap60_2	0,002	0,272	0,007	0,011	ap170_2	0,032	0,825	0,179	0,291	BR6_402_0.2	0,358	0,695	2,824	3,381
ap60_4	0,045	0,27	0,01	0,012	ap170_4	0,07	0,753	0,219	0,288	BR6_402_0.4	0,3	0,403	2,554	3,299
ap60_6	0,007	0,284	0,006	0,011	ap170_6	0,034	0,823	0,185	0,238	BR6_402_0.6	0,337	0,499	2,627	3,381
ap60_8	0,028	0,284	0,007	0,014	ap170_8	0,048	0,843	0,187	0,24	BR7_1000_0.2	6,519	7,281	55,591	62,126
ap70_2	0,049	0,319	0,013	0,019	ap200_2	0,094	0,242	0,306	0,389	BR7_1000_0.4	6,153	9,765	52,511	64,321
ap70_4	0,009	0,316	0,014	0,015	ap200_4	0,092	0,224	0,302	0,388	BR7_1000_0.6	5,572	8,134	50,269	64,03
ap70_6	0,029	0,328	0,013	0,019	ap200_6	0,055	0,217	0,321	0,39	BR7_1000_0.8	6,676	8,665	56,526	63,017
ap70_8	0,022	0,336	0,011	0,02	ap200_8	0,068	0,219	0,304	0,4	BR8_2000_0.2	64,245	86,699	686,325	784,339

Continuação Tabela 2: Tempo médio em segundos para criação de uma solução utilizando os métodos de construção paralelo e sequencial.

Instância	$M1^p$	$M2^p$	$M1^s$	$M2^s$
BR8_2000_0.4	63,77	86,69	686,32	784,33
BR8_2000_0.6	65,71	94,54	702,97	728,26
BR8_2000_0.8	74,35	93,41	798	726,68
BR9_3038_0.2	227	232,47	2110,05	2069
BR9_3038_0.4	291,4	236,37	2430,45	1983,87
BR9_3038_0.6	243,98	238,42	2151,41	2000,22
BR9_3038_0.8	242,37	243,83	2137,21	2003,21

### 4.3.2 Influência do número de *threads* no comportamento do algoritmo ILS-RVND1

Uma análise para verificar o impacto no desempenho do método proposto, em relação ao número de *threads* e *hardware* utilizados, é feita nesta seção. Sendo as variantes do método ILS-RVND1, com diferentes quantidades de *threads*, nomeadas como  $H_t$ , onde  $t$  é o número de *threads* utilizadas no experimento. Foram comparados seis diferentes possibilidades: H1, H6, H10, H18, H34 e H66. O objetivo é avaliar o desempenho em relação ao indicador *Speedup* em diferentes configurações de computador. O experimento consiste em executar o algoritmo sequencial 10 vezes durante  $0.4n$  segundos, e posteriormente contabilizar o tempo necessário para as diferentes configurações do algoritmo paralelo obter, em média, a mesma solução. A Tabela 4 mostra os resultados médios de *Speedup* obtidos.

Os resultados mostram que o desempenho do algoritmo piora a medida que o número de *threads* utilizados ultrapassa o número de *threads* disponíveis no processador. Este comportamento pode ser explicado devido ao grande número de *threads*, além das disponibilizadas pelo CPU, implicar na contenção de recursos por *threads*, reduzindo a eficiência paralela. Por exemplo, elas poderiam ficar disputando tempo de CPU, ficando ociosas por um tempo significativo.

Em relação ao *hardware*, o maior poder de processamento, junto ao número ideal de *threads* disponibilizadas ao algoritmo, resulta no alcance de soluções alvo de forma mais rápida (maior *speedup*). Neste trabalho é chamado de número ideal de *threads* a situação onde  $t$  está próximo do número de *threads* disponíveis no *hardware*. Embora não seja observado um ganho linear de eficiência com aumento de recursos, mais especificamente, número de *threads* disponíveis, foram obtidos ganhos significativos.

Ainda vale ressaltar que o método de busca, ILS-RVND1, não tem como objetivo principal ser mais rápido, ao contrário do que acontece com o método de construção, que paralelizando a nível de iteração se tem o mesmo resultado em menor tempo. O principal objetivo aqui é explorar de forma mais eficiente o espaço de busca para encontrar as

Tabela 3 – Valor de função objetivo obtido pelos métodos de construção 1 e 2

Instancia	$MC^S$	$MC^L$	Instancia	$MC^S$	$MC^L$
ap10_2	91132192,78	91132192,78	ap150_2	58269485,48	57562102,89
ap10_4	95161467,58	95161467,58	ap150_4	69894125,84	68644646,87
ap10_6	95161467,58	95161467,58	ap150_6	80320679,04	79004087,64
ap10_8	95161467,58	95161467,58	ap150_8	90045148,80	88810725,63
ap20_2	93546566,22	91624495,16	ap170_2	65379318,24	64008524,09
ap20_4	96673177,86	98396315,22	ap170_4	75365531,17	74010849,59
ap20_6	98181949,71	102365427,49	ap170_6	86879344,06	83145529,91
ap20_8	98181949,71	98181949,71	ap170_8	92716214,70	89973180,31
ap30_2	84275516,21	83669102,06	ap200_2	64635671,47	62878552,86
ap30_4	91688917,22	92164610,92	ap200_4	74110727,34	73141447,50
ap30_6	95631853,67	97162185,33	ap200_6	85153199,81	82925933,02
ap30_8	99454677,78	99454677,78	ap200_8	92095146,96	90956576,34
ap40_2	81855577,94	80537210,15	BR1_100_0.2	1463,61	1463,61
ap40_4	89079325,86	88841173,59	BR1_100_0.4	1693,51	1693,51
ap40_6	95565068,67	95565068,67	BR1_100_0.6	1917,50	1917,50
ap40_8	99892619,69	99892619,69	BR1_100_0.8	2088,55	2088,55
ap50_2	73625312,11	72279229,79	BR2_200_0.2	4212,24	4036,51
ap50_4	83364128,97	81743988,20	BR2_200_0.4	4844,57	4680,70
ap50_6	91170489,24	90789884,73	BR2_200_0.6	5370,95	5341,71
ap50_8	97189855,57	98740092,05	BR2_200_0.8	5805,38	5777,47
ap60_2	64937566,19	65013185,77	BR3_300_0.2	6517,69	6034,84
ap60_4	73737884,34	73724618,82	BR3_300_0.4	7638,39	7216,97
ap60_6	81791079,47	81791079,47	BR3_300_0.6	8916,71	8312,11
ap60_8	87771774,70	87771774,70	BR3_300_0.8	9851,34	9311,78
ap70_2	76773268,75	74957102,30	BR4_300_0.2	6686,24	6332,03
ap70_4	85031871,83	84077732,93	BR4_300_0.4	7863,72	7503,40
ap70_6	93254204,48	92977367,92	BR4_300_0.6	9069,92	8612,05
ap70_8	98821238,41	100675260,42	BR4_300_0.8	9991,83	9720,70
ap80_2	75678463,94	71173512,02	BR5_402_0.2	8468,54	8378,04
ap80_4	84587197,52	80475541,11	BR5_402_0.4	10166,11	10042,59
ap80_6	93450021,47	90073150,17	BR5_402_0.6	11741,31	11647,55
ap80_8	98790990,68	99670759,23	BR5_402_0.8	13179,65	12992,56
ap90_2	72837815,82	69319095,13	BR6_402_0.2	8787,69	8460,42
ap90_4	82047115,74	80433713,19	BR6_402_0.4	10360,37	10133,84
ap90_6	88486437,12	91124489,08	BR6_402_0.6	11879,35	11624,74
ap90_8	93724802,56	96719584,09	BR7_1000_0.2	116,83	115,56
ap100_2	70721826,44	67609438,70	BR7_1000_0.4	150,02	148,46
ap100_4	80344130,52	78580542,74	BR7_1000_0.6	182,00	179,94
ap100_6	88561748,58	87501795,76	BR7_1000_0.8	210,04	210,44
ap100_8	94230505,66	96293624,10	BR8_2000_0.2	227,19	225,71
ap130_2	67553025,87	60922955,13	BR8_2000_0.4	298,49	294,70
ap130_4	77910334,91	71788815,61	BR8_2000_0.6	363,20	359,36
ap130_6	87837006,08	82317281,11	BR8_2000_0.8	420,17	417,84
ap130_8	94780030,14	92045310,05	BR9_3038_0.2	54249,27	53531,05

Continuação Tabela 3: Valor de função objetivo obtido pelos métodos de construção 1 e 2.

Instancia	$MC^S$	$MC^L$	Instancia	$MC^S$	$MC^L$
BR9_3038_0.4	73623,91	72356,81	BR9_3038_0.8	108604,21	108368,61
BR9_3038_0.6	91939,63	90590,62			

melhores soluções. Se o algoritmo faz isso de forma eficiente, naturalmente a probabilidade de encontrar estas soluções de forma mais rápida que o sequencial, aumenta. O algoritmo ILS-RVND1 será configurado com  $t = 18$  por apresentar os melhores resultados.

Tabela 4 – Speedup obtido com diferentes configurações do algoritmo ILS-RVND1 e *hardware*

Indicador	H1	H6	H10	H18	H34	H66
PC1	-	2,80	2,12	1,69	1,24	0,66
PC2	-	3,99	3,60	2,93	2,35	1,58
PC3	-	4,57	7,32	8,14	7,28	3,49

### 4.3.3 Influência do número de *threads* no comportamento do algoritmo ILS-RVND2

No experimento anterior já foi mostrado que o poder de processamento do *hardware* influencia no resulta obtido pelo algoritmo paralelo. Assim sendo, nesta seção os esforços serão concentrados a verificar o número "ideal" de *threads* de busca no computador *PC3* para o método *ILSRVND2*. Foram testadas sete diferentes configurações deste algoritmo, sendo elas *H1*, *H4*, *H5*, *H6*, *H7*, *H8*, *H9* e *H10*. O experimento consiste em executar o algoritmo com 1 *thread* de busca 10 vezes durante  $0.4n$  segundos, e posteriormente, utilizar o valor médio das soluções obtidas como alvo para as outras versões. Assim, será contabilizado o tempo necessário para as diferentes configurações do algoritmo paralelo obter, em média, a mesma solução. A Tabela 5 mostra os resultados médios de *Speedup* para cada variante.

Tabela 5 – *speedup ILSRVND2*

H4	H5	H6	H7	H8	H9	H10
1,09	1,11	1,15	1,19	1,31	1,03	1,01

Como pode ser observado, o incremento no número de *threads* não aumenta de forma significativa o *speedup*, isto pode ser explicado devido ao fato das vizinhanças serem exploradas paralelamente, o que consome todos os núcleos disponíveis. Ainda, vale ressaltar que a versão *H1* deste método utiliza as vizinhanças de forma paralelizada, o objetivo aqui é definir quantas *threads* de busca devem ser utilizadas. Assim sendo, no restante

deste trabalho será utilizado, para o ILS-RVND2,  $t=8$ , por apresentar melhores resultados. O *speed up* médio, em relação ao algoritmo sequencial, para  $t=8$ , é de 12,3.

#### 4.3.4 Análise dos componentes do ILS-RVND1 e ILS-RVND2

Nesta seção são apresentadas algumas conclusões referentes ao impacto dos componentes do ILS-RVND-PR na convergência e na qualidade da solução. O primeiro teste consistiu em alterar o método de criação de uma solução, trocando para um método que gera soluções pseudo aleatórias. Neste caso, foi controlado o número de concentradores instalados inicialmente em  $\lfloor 1 + 0.05n \rfloor$ . Os resultados obtidos mostraram que o algoritmo continua convergindo, na maioria das vezes, para as mesmas soluções, porém aumenta significativamente o tempo de convergência.

Ao analisar as estruturas de vizinhanças, a partir dos experimentos executados, foi verificado que algumas são essenciais para o funcionamento do algoritmo. Os movimentos de "Troca de Função" e "Realocação" tem um alto impacto na qualidade das soluções. Ao se retirar os movimentos "Adicionar e Remover Concentradores", contabilizamos um menor impacto na convergência de ambos os métodos. Não menos importante, ao paralelizar a busca, foi percebido que a convergência do ILS-RVND2 para algumas soluções, em relação ao ILS-RVND1, foi consideravelmente mais rápida. Porém, para algumas instâncias, este método não convergiu, o que também pode ser explicado devido ao fato do número de *threads* de busca ser menor. Com menos cooperação e sem a segmentação das soluções para aplicação da perturbação, a convergência foi prejudicada. Para problemas de larga escala, onde  $n > 1000$ , o ILS-RVND2 obteve melhores resultados, isto pode ser explicado porque a exploração da vizinhança de forma paralela permitiu este algoritmo explorar mais soluções que ILS-RVND1.

Já o método *Path-Relinking* (PR), permite alcançar algumas soluções de qualidade mais facilmente. Verificamos que seu uso foi mais efetivo em instâncias onde  $n < 1000$ . Em alguns casos, sem este recurso o algoritmo não convergiu para a mesma solução. Em outros, o tempo de convergência foi até 4 vezes maior. Por exemplo, a convergência para a solução ótima da instância ap150\_8, no intervalo de tempo definido neste trabalho, é comprometida ao retirar o PR. Em 10 execuções o algoritmo converge apenas 5 vezes sem este recurso.

Finalmente, os procedimentos de perturbação permitiram o algoritmo sair de ótimos locais. Ao retirar a segmentação dos vetores soluções para aplicar a perturbação, entre as diferentes *threads*, o tempo de convergência de ambos os algoritmos aumentou. Além disso, altos valores iniciais para  $\pi$  também impactam negativamente nos algoritmos. Assim sendo, é entendido que todos os recursos utilizados no algoritmo são necessários para o seu funcionamento.

### 4.3.5 Proposta $\times$ Literatura

Nesta seção, serão avaliadas as heurísticas proposta neste trabalho em relação a heurística considerada o atual estado da arte para o problema e um método exato proposto por [Gomes \(2016\)](#). O método baseado na meta-heurística Busca Tabu apresentado por [Abyazi-Sani e Ghanbari \(2016\)](#) foi escolhido por apresentar, entre os principais algoritmos da literatura, o melhor resultado, como pode ser visto em [Carvalho et al. \(2017\)](#).

Porém, este algoritmo utiliza o método  $M2^s$  de construção, e como pode ser visto na seção onde foi verificado a eficiência dos métodos de construção, a abordagem sequencial demanda tempo significativamente superior quando comparado ao paralelo, especialmente quando se trata de problemas de grande dimensão. Assim sendo, o método de construção sequencial do Busca Tabu foi substituído pelo mesmo método paralelo, o que não acarreta modificações no comportamento do algoritmo, apenas uma aceleração neste procedimento. Em relação a métodos exatos aplicado a este problema, a Decomposição de Benders tem se destacado. Aqui será utilizado um dos métodos de Decomposição proposto por [Gomes \(2016\)](#), por apresentar bons resultados. Em relação aos experimentos, duas questões fundamentais, na concepção de métodos heurísticos, serão avaliadas: eficiência, em relação a tempo computacional para obtenção de soluções de qualidade; e qualidade das soluções encontradas.

Inicialmente foram computados os seguintes indicadores: KBS, DevMin e DevMed, explicados no capítulo 2. Como critério de parada para os algoritmos, foi utilizado o tempo de máquina limitado em aproximadamente  $0,4n$  segundos. Este tempo foi definido por se tratar da maior proporção de tempo de execução, em relação a  $n$ , reportado por [Abyazi-Sani e Ghanbari \(2016\)](#) para obter soluções de qualidade. A Tabela 6 mostra os valores dos indicadores obtidos neste trabalho, para os três métodos heurísticos, após 10 execuções de cada método para as 91 instâncias.

Tabela 6 – Comparação entre os métodos heurísticos

Indicadores	ILS-RVND2	ILS-RVND1	<a href="#">Abyazi-Sani e Ghanbari (2016)</a>
KBS	852	818	740
DevMin	0,0001	0,0001	0,099
DevMed	0,0002	0,0003	0,099

Considerando as 910 replicações, o ILS-RVND2 encontrou soluções melhores ou iguais as encontradas pelos os outros dois métodos em aproximadamente 94% das replicações. Pelos valores de DevMin e DevMed, é possível perceber que ambos os métodos possuem um bom comportamento, ou seja, convergem sempre para soluções de qualidade. A tabela 7 mostra os valores médio de função objetivo obtidos neste experimento. Valores

diferenciados de função objetivo foram destacados, nestes casos, um dos algoritmos foi melhor em relação a pelo menos um dos outros dois.

Os resultados mostram o quanto são eficientes os algoritmos propostos neste trabalho, bem como a heurística considerada o atual estado da arte. É importante ressaltar que o comportamento do Busca Tabu descrito neste trabalho é melhor do que aquele apresentado na literatura. A paralelização na construção permitiu um ganho significativo de tempo de convergência. Por exemplo, na Figura 30, é possível verificar que o tempo de convergência para instância BR8\_2000\_8 é de aproximadamente 300 segundos. Caso fosse utilizado o método sequencial para construção da solução inicial, este mesmo algoritmo demandaria, em média, 836 segundos para obter a mesma solução. Isso significa uma redução de 65% do tempo. Isso reforça a ideia que o uso da paralelização pode contribuir para resolver problemas de grande escala. Com uma pequena modificação deste método foi possível obter ganhos significativos.

Ao analisar os indicadores e os valores de função objetivo temos bons indícios que os algoritmos propostos neste trabalho são superiores, pois apresentam os melhores resultados. Se tratando apenas dos métodos ILS-RVND1 e ILS-RVND2, os resultados mostram que para instâncias onde  $n \leq 1000$ , o algoritmo ILS-RVND1 é mais eficiente. Este comportamento muda para instâncias de maior porte, onde  $n \geq 2000$ . Neste cenário, esta abordagem não demonstrou tanta eficiência, pois apresentou resultados piores para função objetivo. Um dos motivos que pode ter favorecido este comportamento é a complexidade/custo computacional para explorar as vizinhanças. Como o ILS-RVND2 explora paralelamente tais estrutura, foram obtidos ganhos de desempenho significativo, acelerando a convergência do mesmo. Por outro lado, ILS-RVND1 se mostrou mais eficiente para explorar o espaço de busca para problemas de até 400 nós, onde obteve 2 soluções melhores que ILS-RVND2.

Ainda, em relação a exploração das vizinhanças, para entender a contribuição da exploração paralela, frente a sequencial, foram feitos testes aferindo o tempo gasto pela busca local utilizando as diferentes estruturas de vizinhança disponíveis. Os resultados mostraram que a exploração de forma paralela foi, no melhor caso, 10 vezes mais rápida. Por exemplo, para instâncias de grande porte ( $n > 1000$ ), enquanto a versão sequencial gastou 58 segundos para explorar, em uma determinada iteração do algoritmo, a vizinhança "troca de função", a versão paralela gastou aproximadamente 6 segundos.

Além disso, para entender o comportamento das heurísticas ILS-RVND1, ILSRVND2 e Busca Tabu, em relação a convergência e tempo de execução, foram feitos gráficos *Time-to-Target* (TTT), metodologia introduzida por Alex, Resende e Ribeiro (2002) e Alex, Resende e Ribeiro (2007). Para uma dada instância, um gráfico TTT exhibe no eixo das coordenadas a probabilidade do algoritmo obter a solução alvo dentro de um determinado tempo de execução (em segundos), mostrado no eixo das abcissas. As instâncias foram escolhidas aleatoriamente, sendo controlado apenas o número de nós. O objetivo foi

amostrar a convergência de diferentes instâncias, em relação ao número de nós. Assim, para cada teste, uma instância de tamanho  $n = \{50, 80, 130, 150, 200, 300, 1000, 2000\}$  foi escolhida. Posteriormente, os tempos necessários para encontrar as soluções alvo foram registrados, para cada instância separadamente, em 200 execuções de cada método. Como solução alvo foi definido o valor médio de função objetivo obtido pelo algoritmo que obteve o pior desempenho na instância selecionada.

As Figuras 29 e 30 mostram os resultados obtidos, considere  $P(\text{Alg1} < \text{Alg2})$  sendo a probabilidade de um algoritmo Alg1 encontrar uma solução mais rápido do que um algoritmo Alg2. Por exemplo, é possível observar que  $P(\text{ILS-RVND1} < \text{Busca Tabu}) = 90\%$  na instância ap10\_2. Também é possível observar que  $P(\text{Busca Tabu} < \text{ILS-RVND1}) = 0$  na instância BR3\_300\_4. Para a instância ap150\_8 o comportamento do algoritmo ILS-RVND1 é atípico, sendo  $P(\text{ILS-RVND1} < \text{Busca Tabu}) = 9\%$ . Embora a convergência seja mais lenta, foi possível alcançar uma solução melhor que a obtida pelo Busca Tabu (ver tabela 7).

Os valores de função objetivo, bem como os gráficos TTT, mostram a degradação do desempenho do ILS-RVND1 a medida que o tamanho das instâncias aumentam. Também é possível observar que o método ILS-RVND2 se mantém competitivo, alcançando as melhores soluções nos problemas de grande escala ( $n \geq 1000$ ). Além disso, mantém alta a probabilidade de encontrar soluções de qualidade de forma rápida. Embora exista a perda de desempenho, quando comparado com o Busca Tabu, ILS-RVND1 ainda é competitivo pois alcança boas soluções, embora demande mais tempo.

É importante lembrar que os resultados obtidos neste experimento, bem como os gráficos que estão sendo mostrados, levam em consideração que o tempo disponível de processamento está limitado a  $0.4n$  segundos. Outros limites de tempo podem mudar os resultados obtidos. Por exemplo, ao aumentar o tempo de processamento, o método ILS-RVND2 encontra a mesma solução encontrada pelo ILS-RVND1 na instância BR5\_402\_0.6. Outras situações podem acarretar uma nova solução alvo para o TTT, e uma nova interpretação em relação a convergência dos algoritmos.

Embora os resultados apresentados forneçam um bom indicador de desempenho, não podem ser utilizados para obter conclusões mais gerais. A avaliação do desempenho foi realizada utilizando o teste estatístico de Friedman (1937), que objetiva identificar se há diferença estatisticamente significativa entre os algoritmos. Testes estatísticos são usados para determinar se as diferenças observadas são reais ou são devido ao acaso. Considerando o nível de significância de 0,05, verificou-se que há uma diferença significativa entre os algoritmos ( $p = 0.0021$ ).

Tabela 7 – Valor de função objetivo obtido pelas diferente heurísticas após  $0,4n$  segundos

Instância	ILS-RVND1	ILS-RVND2	Busca Tabu	Instância	ILS-RVND1	ILS-RVND2	Busca Tabu
ap10_2	90963539,48	90963539,48	90963539,48	ap80_2	70713485,99	70713485,99	70713485,99
ap10_4	95079629,91	95079629,91	95079629,91	ap80_4	79704787,95	79704787,95	79704787,95
ap10_6	95161467,58	95161467,58	95161467,58	ap80_6	88418089,75	88418089,75	88418089,75
ap10_8	95161467,58	95161467,58	95161467,58	ap80_8	95798238,88	95798238,88	95798238,88
ap20_2	91507336,61	91507336,61	91507336,61	ap90_2	69223173,92	69223173,92	69223173,92
ap20_4	96673177,86	96673177,86	96673177,86	ap90_4	78931357,81	78931357,81	78931357,81
ap20_6	98181949,71	98181949,71	98181949,71	ap90_6	87012179,95	87012179,95	87012179,95
ap20_8	98181949,71	98181949,71	98181949,71	ap90_8	92780846,25	92780846,25	92780846,25
ap30_2	83576754,71	83576754,71	83576754,71	ap100_2	67584119,76	67584119,76	67584119,76
ap30_4	91209293,10	91209293,10	91209293,10	ap100_4	77545112,32	77545112,32	77545112,32
ap30_6	95053569,42	95053569,42	95053569,42	ap100_6	86371515,86	86371515,86	86371515,86
ap30_8	98520500,58	98520500,58	98520500,58	ap100_8	93184508,97	93184508,97	93184508,97
ap40_2	80537210,15	80537210,15	80537210,15	ap130_2	60901562,49	60901562,49	60901562,49
ap40_4	88042046,09	88042046,09	88042046,09	ap130_4	71560028,43	71560028,43	71560028,43
ap40_6	94523086,45	94523086,45	94523086,45	ap130_6	81332451,63	81332451,63	81332451,63
ap40_8	99180263,10	99180263,10	99180263,10	ap130_8	89960079,07	89960079,07	89960079,07
ap50_2	71261044,61	71261044,61	71261044,61	ap150_2	57397001,45	57397001,45	57397001,45
ap50_4	80325464,90	80325464,90	80325464,90	ap150_4	<b>68081635,57</b>	<b>68081635,57</b>	68133832,84
ap50_6	89389885,19	89389885,19	89389885,19	ap150_6	77856856,14	77856856,14	77935631,22
ap50_8	95205946,96	95205946,96	95205946,96	ap150_8	<b>86068900,60</b>	<b>86068900,60</b>	86113865,55
ap60_2	64790967,04	64790967,04	64790967,04	ap170_2	63888877,91	63888877,91	63888877,91
ap60_4	73074656,58	73074656,58	73074656,58	ap170_4	73373712,12	73373712,12	73373712,12
ap60_6	80673823,66	80673823,66	80673823,66	ap170_6	<b>81744592,07</b>	<b>81744592,07</b>	81915993,39
ap60_8	87285162,09	87285162,09	87285162,09	ap170_8	88839517,44	88839517,44	88839517,44
ap70_2	74451085,60	74451085,60	74451085,60	ap200_2	62774425,09	62774425,09	62774425,09
ap70_4	83272519,63	83272519,63	83272519,63	ap200_4	72333785,14	72333785,14	72333785,14
ap70_6	91741977,20	91741977,20	91741977,20	ap200_6	81236269,73	81236269,73	81236269,73
ap70_8	<b>97109841,98</b>	97177453,52	<b>97109841,98</b>	ap200_8	89181926,79	89181926,79	89181926,79

Continuação Tabela 7: Valor de função objetivo obtido pelas diferentes heurísticas após  $0,4n$  segundos

Instância	ILS-RVND1	ILS-RVND2	Busca Tabu	Instância	ILS-RVND1	ILS-RVND2	Busca Tabu
BR1_100_0.2	1455,37	1455,37	1455,37	BR7_1000_0.2	<b>115,15</b>	<b>115,15</b>	115,19
BR1_100_0.4	1682,03	1682,03	1682,03	BR7_1000_0.4	147,44	<b>147,32</b>	147,48
BR1_100_0.6	1890,08	1890,08	1890,08	BR7_1000_0.6	<b>178,20</b>	178,27	178,21
BR1_100_0.8	2081,36	2081,36	2081,36	BR7_1000_0.8	<b>207,32</b>	207,54	207,78
BR2_200_0.2	4036,01	4036,01	4036,01	BR8_2000_0.2	224,54	<b>224,41</b>	224,88
BR2_200_0.4	4677,49	4677,49	4677,49	BR8_2000_0.4	290,58	<b>290,31</b>	291,29
BR2_200_0.6	5255,39	5255,39	5255,39	BR8_2000_0.6	355,8	<b>354,85</b>	356,01
BR2_200_0.8	5654,95	5654,95	5654,95	BR8_2000_0.8	414,01	<b>413,00</b>	413,55
BR3_300_0.2	6024,08	6024,08	6024,08	BR9_3038_0.2	53408,27	<b>53311,23</b>	53372,42
BR3_300_0.4	7173,19	7173,19	7173,19	BR9_3038_0.4	72074,67	<b>71833,17</b>	71893,68
BR3_300_0.6	8229,17	8229,17	8229,17	BR9_3038_0.6	90071,48	<b>89607,62</b>	89994,84
BR3_300_0.8	<b>9205,23</b>	<b>9205,23</b>	9205,40	BR9_3038_0.8	107345,65	<b>106792,54</b>	1066362,77
BR4_300_0.2	6329,22	6329,22	6329,22				
BR4_300_0.4	7483,56	7483,56	7483,56				
BR4_300_0.6	8562,62	8562,62	8562,62				
BR4_300_0.8	9589,02	9589,02	9589,02				
BR5_402_0.2	8357,49	8357,49	8357,49				
BR5_402_0.4	9961,60	9961,60	9961,60				
BR5_402_0.6	<b>11499,28</b>	11506,14	11506,14				
BR5_402_0.8	<b>12906,98</b>	<b>12906,98</b>	12907,01				
BR6_402_0.2	8432,65	8432,65	8432,65				
BR6_402_0.4	<b>10083,32</b>	<b>10083,32</b>	10084,80				
BR6_402_0.6	11546,69	11546,69	11546,69				

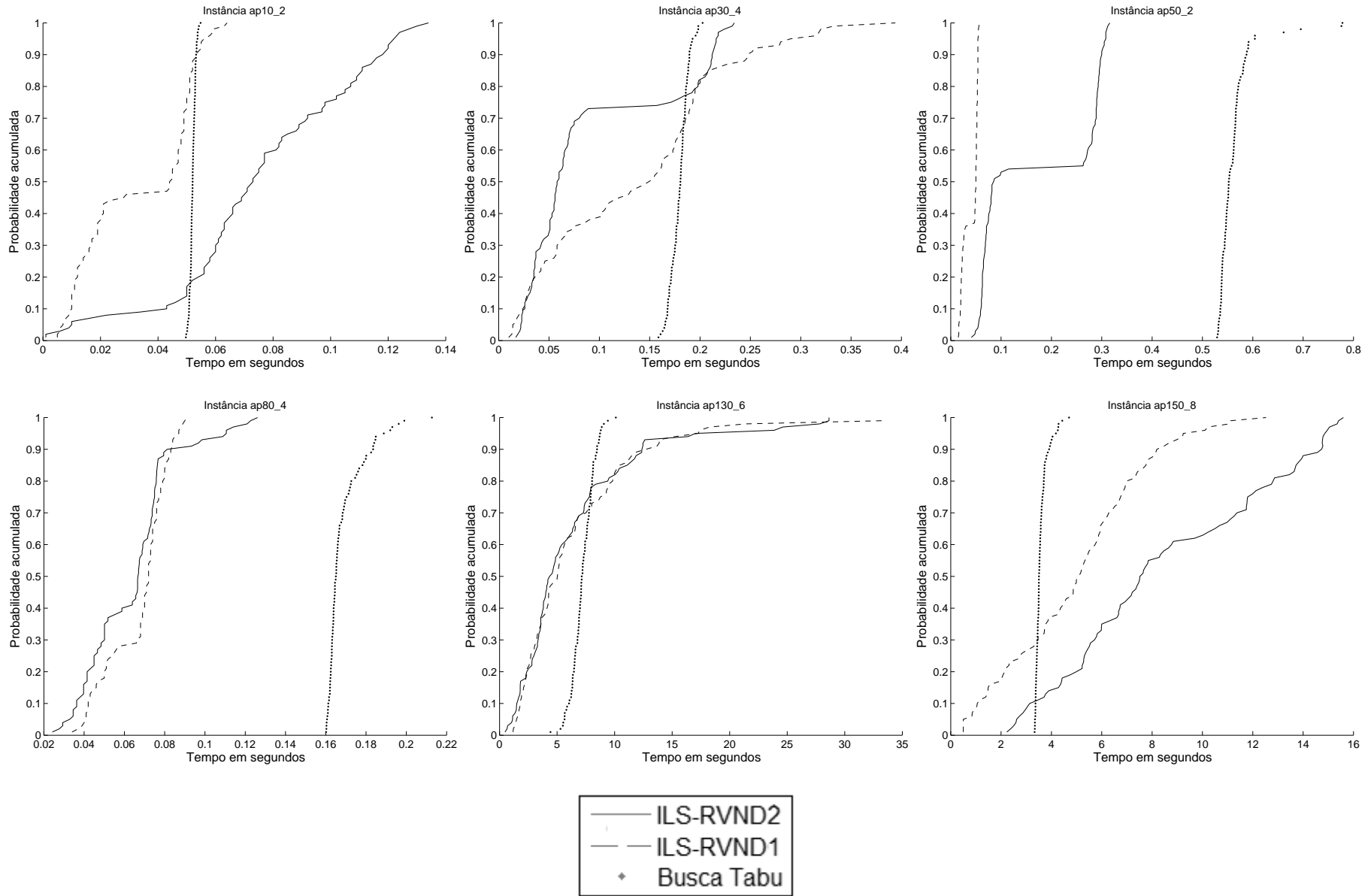


Figura 29 – Gráficos TTT.

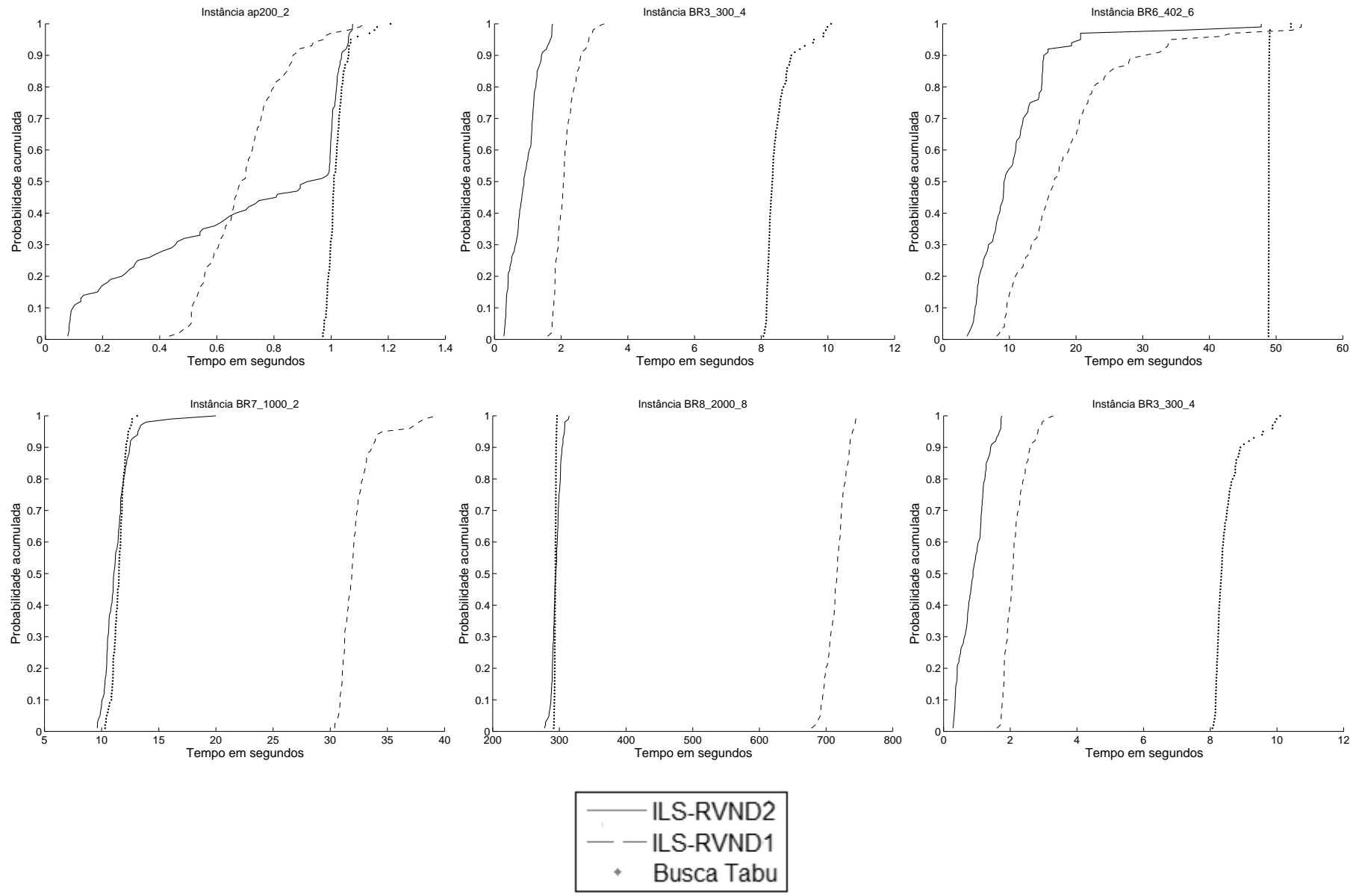


Figura 30 – Gráficos TTT.

Além disso, constatou-se a partir do método *multcompare* (MATLAB) que os métodos propostos superam a heurística da literatura. No teste de Friedman, quando a hipótese nula  $H_0$  é rejeitada, temos que, ao menos um dos grupos é diferente dos demais. Porém, não temos a informação de quais grupos são diferentes. Neste sentido, o procedimento de comparações múltiplas (*multcompare*) nos permite determinar quais grupos são diferentes. Esse procedimento retorna um gráfico que pode ser interpretado da seguinte forma: duas médias são significativamente diferentes se os intervalos são disjuntos, e não são significativamente diferentes, se os seus intervalos se sobrepõem. Quanto mais à esquerda a média e o intervalo de confiança, melhor o algoritmo. Isto porque esta análise é baseada em *rank*, e quanto melhor o valor de função objetivo para uma determinada instância, menor é o valor do *rank* associado. Assim, pode-se concluir que as diferentes propostas deste trabalho não são significantemente diferentes. Porém, ambas são diferentes e melhores em relação ao atual estado da arte.

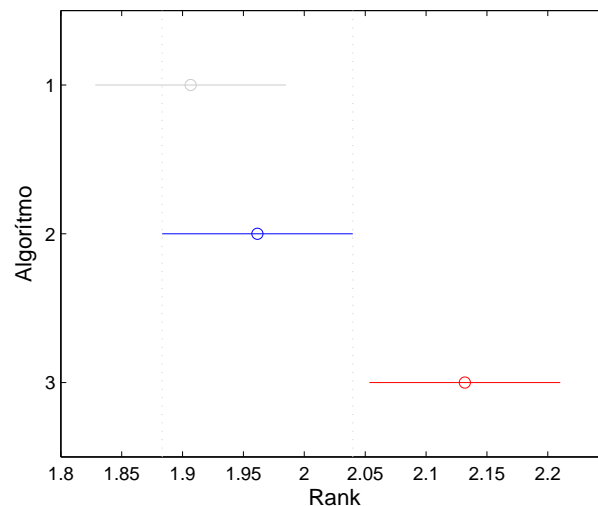


Figura 31 – Gráfico contendo informação de média e desvio padrão em relação ao Rank, obtido a partir do teste de Friedman e do procedimento de comparações múltiplas. Algoritmos: 1 - ILS-RVND2; 2 - ILS-RVND1; 3 - Busca Tabu.

Quando comparado com os valores obtidos pelo método de Decomposição de Benders, para instâncias de até 200 nós, o método ILS-RVND1 alcançou o ótimo em todos os casos. O método ILS-RVND2, não alcançou em uma das instâncias, ap70\_8, ficando a aproximadamente 1% da solução ótima. Já o método Busca Tabu, não alcançou tais soluções em três instâncias: ap150\_6, ap150\_8 e ap170\_6. Nestes casos o algoritmo ficou preso em ótimos locais e mesmo aumentando o limite de tempo (ou número de iterações) não houve melhora na função objetivo.

Vale ressaltar que para obter os valores ótimos destas instâncias foram necessário até 39000 segundos de processamento do método de Benders. Para instâncias consideradas de grande porte, onde  $n \geq 500$  se torna impraticável a utilização desta metodologia, devido

o tempo computacional demandado. Vale ressaltar que esta afirmação é verdadeira para o algoritmo de Benders implementado neste trabalho.

## 4.4 Conclusão

O projeto de redes do tipo E-R não capacitado com alocação simples foi abordado neste capítulo. Para resolução deste problema foram propostas duas heurísticas paralelas, denominadas ILS-RVND1 e ILS-RVND2.

Inicialmente, foram apresentadas duas ideias relacionadas aos principais métodos de construção utilizados para o problema de planejamento de redes do tipo E-R não capacitado com alocação simples. O comportamento destes métodos foi estudado, além disso, versões paralelas foram propostas com o objetivo de obter soluções de qualidade em um espaço de tempo significativamente menor. Os resultados apontaram a superioridade de umas das versões, bem como a eficiência da utilização da paralelização para ambos os métodos.

Para refinar as soluções geradas foram propostas duas heurísticas baseadas nos métodos ILS, RVND e *Path-Relinking*. Tais heurísticas diferem-se por utilizar o recurso de paralelização de forma distinta. A primeira, ILS-RVND1, utiliza várias buscas em paralelo, cada uma trabalhando de forma sequencial. Já o método ILS-RVND2, utiliza menos *threads* de busca, porém cada uma explora a vizinhança paralelamente.

Os experimentos feitos neste trabalho mostram que as duas propostas superaram o atual estado da arte e tiveram comportamento semelhantes. Embora o teste estatístico tenha mostrado que as versões não apresentam diferença estatística significativa, é possível perceber que se levado em consideração apenas problemas de grande porte, onde  $n > 1000$ , a versão denominada ILS-RVND2 obteve melhores resultados. Isso porque a exploração das vizinhanças de forma sequencial para problemas de grande escala se mostrou menos eficiente, dentro do tempo disponibilizado para os testes.

Desta forma, foram evidenciados a viabilidade do uso dos métodos propostos, bem como o benefício que a paralelização pode fornecer, tais como obtenção de melhores soluções ou aceleração dos métodos. Não menos importante, verificamos que a estratégia de busca aplicada pelo método Busca Tabu é muito eficiente. Por outro lado, o método proposto neste trabalho e de simples implementação e obtém resultados melhores.



## 5 Redes do tipo Eixo-Raio com alocação múltipla

Neste capítulo será introduzido o problema de redes do tipo E-R com alocação múltipla, custos fixos e não capacitado. Além disso, uma revisão bibliográfica listando os principais métodos para resolver o problema, bem como as metodologias propostas são apresentadas. Serão detalhados a representação computacional de uma solução, função de avaliação, o método de construção, perturbações, os movimentos utilizados para a busca local e a estratégia adotada para o método *Path-Relinking*. Finalmente, os resultados e conclusões obtidas são apresentados.

### 5.1 Introdução

Esse problema pode ter diferentes definições, a que será abordada neste trabalho é baseada nas seguintes: Dado  $N$  um conjunto de nós de demanda e  $K$  um conjunto de nós candidatos a concentradores, onde  $n = |N|$ . Vamos considerar que  $K \subseteq N$ , ou seja, todos os nós de demanda são candidatos a concentradores, portanto  $K \equiv N$ . Para todo par de nós  $i$  e  $j$  ( $i, j \in N : i \neq j$ ),  $w_{ij}$  representa o fluxo de demanda entre o nó  $i$  (origem) e o nó  $j$  (destino) que é roteado por um ou dois concentradores. Normalmente  $w_{ij} \neq w_{ji}$ . Ainda, dado  $f_k$  sendo o custo de instalação de um concentrador em um nó  $k \in N$  e  $c_{ijkm}$  o custo de transportar uma unidade de fluxo vindo da origem  $i$  com destino  $j$  sendo roteado pelos concentradores  $k$  e  $m$ , ou seja, o custo da rota  $i - k - m - j$ . O custo de transporte pode ser decomposto em três segmentos:  $c_{ijkm} = c_{ik} + \alpha c_{km} + c_{mj}$ , onde  $c_{ik}$  e  $c_{mj}$  é o custo de transporte do nó  $i$  até o concentrador  $k$  e do concentrador  $m$  até o nó  $j$  respectivamente. Ainda temos  $\alpha c_{km}$  como o custo de transporte entre os concentradores  $k$  e  $m$ , nesse caso o custo é ponderado por um fator  $\alpha$  que visa representar o fator de desconto, ou economia de escala, entre os nós concentradores. Se apenas um concentrador é utilizado, isso significa que  $k = m$ , nenhum fator de desconto é utilizado. Nesse caso consideramos que existe uma aresta ligando todos os concentradores. Além disso, o número de concentradores não é conhecido a priori. A definição deste problema é praticamente a mesma quando comparada com o problema tratado no capítulo anterior, o que difere os problemas é que os fluxos originados em um nó  $i$  para diferentes nós  $j$  podem ser roteados por diferentes concentradores, ou seja, é permitido alocações múltiplas.

Um exemplo de uma rede do tipo Eixo-Raio com alocação múltipla é mostrado na Figura 32, onde os concentradores estão representados como triângulos, os pontos de demanda como círculos e as alocações são representadas pelas linhas, bem como as conexões

entre os concentradores. Observa-se que os pontos de demanda (nós não concentradores) podem estar ligados a mais de um concentrador.



Figura 32 – Exemplo de uma rede do tipo E-R com alocação múltipla

Assim, o objetivo é determinar quais nós devem ser concentradores e para quais concentradores cada fluxo deve ser atribuído/direcionado, de tal forma que os custos totais, transporte e de instalação, sejam minimizados. Uma formulação para este problema pode ser encontrada em Hamacher et al. (2000), conforme as equações (5.1)-(5.3). Considere a variável  $y_i = 1$  caso o nó  $i \in N$  é um concentrador, ou  $y_i = 0$ , caso contrário. Além disso, considere  $x_{ijkm}$  sendo a fração do fluxo  $w_{ij}$  roteado pelos concentradores  $k$  e  $m$ .

$$\min \phi' = \sum_k f_k z_{kk} + \sum_i \sum_j \sum_k \sum_m c_{ijkm} x_{ijkm} \quad (5.1)$$

$$\sum_{m \in N, m \neq k} x_{ijmk} + \sum_{m \in N} x_{ijkm} \leq y_k \quad \forall i, j, k \in N \quad (5.2)$$

$$\sum_{k \in N} \sum_{m \in N} x_{ijkm} = 1 \quad \forall i, j \in N \quad (5.3)$$

$$x_{ijkm} \geq 0 \quad \forall i, j, k, m \in N \quad (5.4)$$

$$y_i \in \{0, 1\} \quad \forall i \in N \quad (5.5)$$

A função objetivo (5.1) busca minimizar os custos de transporte e instalação de concentradores. As restrições (5.2) garantem que o fluxo passa por um nó intermediário somente se ele for concentrador. As restrições (5.3) garantem que o fluxo, para qualquer par de nós  $(i, j)$ , será atendido. Finalmente, as restrições (5.4) e (5.5) definem a integralidade das variáveis.

Existem atualmente na literatura vários trabalhos tratando problemas de localização de concentradores com alocação múltipla. Uma das primeiras formulações matemática

linear foi introduzido por [Campbell \(1994b\)](#). Porém não foram feitos testes computacionais, o que tornou o trabalho apresentado por [Klincewicz \(1996\)](#) o primeiro a propor um modelo e obter soluções de forma exata. Desde então vários trabalhos, tratando diferentes variantes deste problema, tem sido propostos. Dentre eles, muitos consideram que o número de concentradores é previamente conhecidos, como pode ser visto nos trabalhos apresentados por [O’Kelly et al. \(1996\)](#), [Ernst e Krishnamoorthy \(1998a\)](#). Outros consideram que existe capacidade nos concentradores, como pode ser visto no trabalho apresentado por [Aykin \(1994\)](#).

[O’Kelly et al. \(1996\)](#) propuseram um modelo linear para o problema de alocação múltipla sem custo de instalação de concentradores e resolveram com o auxílio do *solver* CPLEX. Os autores variaram o valor da economia de escala para analisar as soluções geradas. Posteriormente, inseriram o custo de instalação com o objetivo de avaliar o impacto deste parâmetro nestas soluções. Além disso, o modelo proposto era composto por menos variáveis e restrições quando comparado aos trabalhos apresentados por [Campbell \(1994b\)](#) e [Skorin-Kapov, Skorin-Kapov e O’Kelly \(1996\)](#). [Ernst e Krishnamoorthy \(1998a\)](#) propuseram formulações para o problema, bem como métodos, heurístico e exato, para resolução do mesmo. Entre as contribuições do trabalho, podemos destacar a possibilidade de resolução de problemas de maior dimensão, quando comparado aos trabalhos da época.

[Hamacher et al. \(2000\)](#) estudaram a estrutura poliédrica deste problema. Desta forma foi possível obter um modelo onde a relaxação linear é justa. [Mayer e Wagner \(2002\)](#) propuseram uma metodologia baseada no método *Branch-and-Bound* para resolver o problema abordado por [Klincewicz \(1996\)](#) de forma mais eficiente. O método utilizado por [Klincewicz \(1996\)](#) foi aperfeiçoado e utilizado no método proposto, com o objetivo de resolver os problemas de forma mais rápida, e obter limites mais precisos. A metodologia foi denominado *Hub Locator*, e obteve, em todos os aspectos, resultados melhores que os apontados por [Klincewicz \(1996\)](#). Porém, ao ser comparado com o *solver* CPLEX, os autores não puderam definir qual metodologia foi mais eficiente. Isso porque em alguns casos o CPLEX supera o *Hub Locator* ou vice-versa.

[Boland et al. \(2004\)](#) utilizaram técnicas específicas de pré processamento a uma formulação baseada na proposta de [Ernst e Krishnamoorthy \(1998a\)](#). Foi possível melhorar os limites inferiores para esta formulação, e ao mesmo tempo diminuir o número de restrições, permitindo assim o uso desta formulação em pacotes computacionais, tal como o CPLEX, para problemas de maiores dimensões. Neste trabalho foram resolvidos problemas com até 50 nós, para algumas instâncias foram necessários aproximadamente 200 horas de processamento.

[Marín \(2005\)](#) apresentou algumas desigualdades válidas que definem facetes para o problema, assim como em [Hamacher et al. \(2000\)](#), desde que os custos associados a distância do nós satisfaça a desigualdade triangular. Os autores citam a importância da

formulação proposta, que dizem ser reduzida e eficiente, mais especificamente com um número menor de variáveis e restrições fortes. Isso permitiu uma redução impressionante nos tempos computacionais e no aumento dos tamanhos das instâncias resolvidas. Os problemas foram resolvidos utilizando o algoritmo *relax-and-cut*, uma classe de algoritmo baseado em Relaxamento Lagrangeano, onde os limites Lagrangeanos são melhorados através do fortalecimento dinâmico das relaxações com a introdução de restrições válidas. Os resultados mostraram a superioridade desta metodologia quando comparado com Boland et al. (2004), embora o número de nós não tenha aumentado em comparação a este trabalho, o tempo foi significativamente menor.

Um algoritmo genético (AG) foi aplicado ao problema de localização de concentradores por Kratica et al. (2005). Foi utilizado a ideia de um algoritmo genético tradicional, e utilizada uma estrutura de dados auxiliar que evita que soluções iguais sejam avaliadas mais de uma vez. Estas soluções (iguais) são processadas de forma que somente uma passa para uma próxima geração. A mutação aplicada neste algoritmo foi direcionada a genes que os autores denominaram como congelados. Estes genes, consistem naquelas posições do vetor solução que possuem um mesmo valor em toda população. Segundo os autores, estes genes não podem ser modificados facilmente pelos operadores de mutação e cruzamento tradicionais, o que diminui consideravelmente a capacidade de exploração destes algoritmos. Para fazer a atribuição dos nós não concentradores aos concentradores, e definir o caminho que o fluxo entre os nós deve passar, foi utilizado o algoritmo Floyd-Warsall modificado, descrito no trabalho apresentado por Ernst e Krishnamoorthy (1998c). Para verificar o desempenho deste algoritmo foram utilizadas instâncias de até 200 nós, sendo que para  $n \leq 120$  foram resolvidos no ótimo. Os valores ótimos foram obtidos em um relatório técnico apresentado por Canovas, Garcia e Marin (2004). Este método foi posteriormente melhorado com a inserção de uma busca local no trabalho apresentado por Djukic (2010). Nesta proposta a cada vez que o algoritmo encontra uma solução melhor que a melhor corrente, uma busca local utilizando o movimento "Troca de função" é aplicada.

Uma heurística baseada nos métodos Busca Tabu e *Simulated Annealing* (SA) foi proposta por Chen (2006). A ideia de combinar estes métodos é devido a boa convergência do SA, e do fato da Busca Tabu utilizar mecanismos que evitam ciclagem. Este método foi dividido em três partes: definir a quantidade de concentradores; definir quem são os concentradores; e definir as rotas para atender as demandas. Inicialmente é utilizada uma heurística para gerar um limite superior para o número de concentradores  $p$ . Posteriormente são escolhidos  $p$  concentradores baseado nos fluxos ( $w$ ) e distâncias ( $c$ ) entre os nós. Como não existe custo na alocação dos nós aos concentradores, é possível considerar que todos os nós não concentradores estão alocados a todos nós concentradores. Assim, o próximo passo foi definir qual a melhor rota para cada  $w_{ij}(i, j \in N)$  a partir de um algoritmo de menor caminho. A cada iteração uma busca local, composta pelo SA e a Busca Tabu, utilizando um movimento que troca os papéis de nós concentradores e não concentradores

é aplicado a solução corrente, bem como a aplicação do algoritmo para obter o menor caminho para atender as demandas. Ao fim da busca local o número de concentradores é incrementado. O método termina quando o número de concentradores atinge o limite  $p$ . A heurística obteve valores ótimos para instâncias de até 50 nós em aproximadamente 2 segundos, significativamente mais rápido que Boland et al. (2004), que, no melhor caso, obtém em aproximadamente 4 horas. Neste trabalho foi resolvido problemas de até 200 nós, porém devido a falta de conhecimento de métodos para comparação, não foi feita nenhuma comparação para  $n > 50$ .

Um algoritmo *Branch-and-Bound* junto a técnica *Dual-Ascent* é proposto por Cánovas, Garcia e Marin (2007). Métodos baseados em *Dual-Ascent* partem de uma solução viável e, a cada iteração, tentam aumentar o valor das variáveis duais desta solução. O método é executado até que não seja mais possível incrementar nenhuma variável dual mantendo a solução dual viável. Pelas condições de complementaridade de folga, é possível construir uma solução primal a partir de qualquer solução dual encontrada por este método. O incremento nos valores das variáveis duais são realizados com o intuito de reduzir o máximo o *Gap* entre as soluções primal e dual. O método proposto foi comparado com o *solver* X-Press, os experimentos mostraram que a proposta foi muito mais eficiente, uma vez que foi possível resolver problemas três vezes maiores que este software ( $n = 120$ ). Vale ressaltar que a metodologia aplicada neste trabalho foi a mesma apresentada por Cánovas, Garcia e Marin (2004).

Diferentes abordagens do algoritmo de Decomposição de Benders foram implementadas por Camargo, Miranda Jr e Luna (2008). Foi utilizada a formulação proposta por Hamacher et al. (2000) por apresentar bons limites. Foi possível resolver problemas de até 200 nós, coisa que até então não era possível com os métodos e ferramentas encontrados na literatura. Os autores sugerem que a utilização de métodos heurísticos para resolver tais problemas já não é mais necessário, devido ao método proposto obter as soluções ótimas em tempo computacional relativamente viável. Isto porque o planejamento e projeto deste tipo de rede cria decisões sobre a aplicação de grande quantidades de recursos, as quais têm considerável impacto nos custos totais de transporte. Assim sendo, utilizar algumas horas de máquina para obter uma solução ótima é justificável. Por outro lado, os autores não consideram problemas onde o número de nós poderiam ser significativamente maiores em relação aos que foram estudados neste trabalho. Isso inviabilizaria o uso do método proposto.

Posteriormente Contreras, Cordeau e Laporte (2011) propuseram uma nova abordagem para o algoritmo de Decomposição de Benders, onde foi possível resolver problemas maiores, com o número de nós chegando a 500. Em relação ao tempo de processamento, quando comparado com o trabalho apresentado por Camargo, Miranda Jr e Luna (2008), este método foi significativamente mais eficiente. Além disso, os autores ressaltam que as

novas instâncias são as maiores e mais difíceis, que até então nenhum trabalho desta área tinha resolvido problemas semelhantes.

Um algoritmo populacional, para problemas de otimização não linear, foi adaptado por Filipovic (2011) para resolver o problema de localização de concentradores com alocação múltipla para  $n \leq 200$ . Este algoritmo é baseado na teoria do eletromagnetismo, mas segue princípios de algoritmos conhecidos, tais como os algoritmos baseados em Enxames e Evolução Diferencial. Embora o artigo não apresente uma comparação com outras heurística, aparentemente os resultados não superam aqueles apresentados por Chen (2006). Por exemplo, na instância com 50 nós o algoritmo não alcança o valor ótimo, enquanto Chen (2006) alcança este valor em aproximadamente 2 segundos.

Como pode ser observado, a maioria dos métodos heurísticos resolvem problemas de até 200 nós. Neste capítulo será analisado o comportamento de algumas desta heurísticas para problemas de grande escala, bem como da metodologia proposta neste trabalho. Vale ressaltar a importância de problemas desta magnitude, que surgem no contexto científico, para testar o desempenho de metodologias, ou em situações práticas, como em problemas logísticos em escala nacional.

## 5.2 Metodologia

Nesta seção será detalhada a heurística proposta para resolver o problema. Vale ressaltar que apenas serão detalhados as partes que compõem a heurística baseada no método ILS-RVND, descrita no capítulo 2.

### 5.2.1 Representação Computacional

Para representar uma solução é utilizado uma lista  $O_s$  contendo quais nós são concentradores. A priori, existem conexões entre todos os nós concentradores e não concentradores. Caso seja necessário recuperar quais conexões estão realmente sendo usadas, é necessário executar um algoritmo para obter o menor caminho entre todos os pontos da rede. Para armazenar os caminhos foi utilizada uma matriz  $M_{n \times n}$  contendo a informação de qual nó adjacente um determinado nó  $i$  deve encaminhar-se para alcançar um determinado nó  $j$  ( $m_{ij}$ ). A Figura 33 ilustra um exemplo desta representação. Neste exemplo, para alcançar o nó 1 partindo do nó 3 é necessário verificar o elemento  $m_{31}$  da matriz. Este indica que o menor caminho entre estes dois nós passa pelo nó 4. Novamente, utilizando o nó 4 como origem, é verificado para qual nó é necessário mover para alcançar o nó de destino, neste caso existe uma conexão até o nó de destino, ou seja,  $m_{41} = 1$ . De forma genérica, o procedimento é repetido até que  $m_{ij} = j$ .

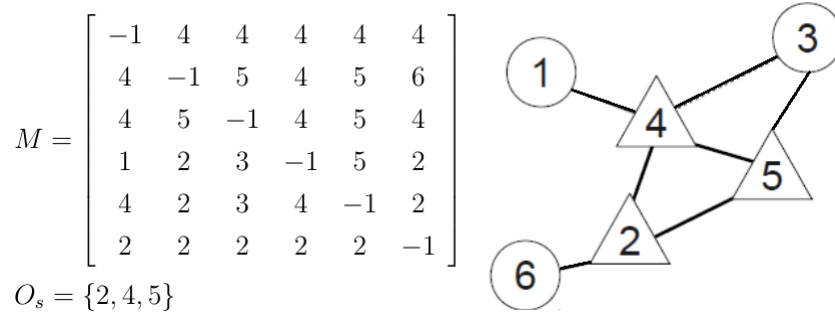


Figura 33 – Representação computacional para o problema de localização de concentradores com alocação múltipla.

### 5.2.2 Avaliação de Solução

Para avaliar uma solução, é utilizado um algoritmo para encontrar o menor caminho entre todos os pontos de um grafo utilizando os concentradores como intermediários. Posteriormente, são calculado os custos de transporte, bem como os custos fixos. Para encontrar o menor caminho entre os pontos foi utilizado um algoritmo baseado no método de Floyd–Warshall (FLOYD, 1962). Para encontrar uma solução, o algoritmo de Floyd–Warshall precisa executar  $n^3$  operações, onde  $n$  é o número de nós do problema. O pseudo-código deste procedimento é mostrado no Algoritmo 4.

---

**Algoritmo 4:** FLOYD-WARSHALL( $c'$ ).

---

```

1 início
2   para  $i = 1$  até  $n$  faça
3     para  $j = 1$  até  $n$  faça
4       para  $k = 1$  até  $n$  faça
5         Se  $c'_{ki} > c'_{kj} + c'_{ji}$  então
6            $c'_{ki} \leftarrow c'_{kj} + c'_{ji}$ 
7         fim
8       fim
9     fim
10  fim
11 fim
```

---

A matriz de custo recebida pelo Algoritmo de Floyd-Warshall segue as seguintes regras:

$$c'_{ij} = \begin{cases} 0, & \text{se } i = j, \\ \infty, & \text{se não existir aresta entre } i \text{ e } j, \\ c_{ij}, & \text{caso contrário.} \end{cases}$$

A cada visita a determinado vértice, os caminhos são recalculados e é verificado se existe um caminho com peso menor para que a matriz  $c'$  seja atualizada.

O algoritmo modificado (ERNST; KRISHNAMOORTHY, 1998b) tem o mesmo objetivo que o algoritmo de Floyd-Warshall, porém utiliza duas etapas para o cálculo do menor caminho. Inicialmente, são calculados os menores caminhos entre os nós concentradores e os não concentradores (diretamente), ou seja,  $d_{kl}$ , onde  $l \in N$  e  $k \in O_s$ . Posteriormente, são calculados os custos entre todos os nós, ou seja,  $c'_{ij}$  para todo  $i, j \in N$ . Este procedimento é ilustrado no Algoritmo 5.

---

**Algoritmo 5:** FLOYD-WARSHALL-MODIFICADO( $c$ )
 

---

```

1 início
2   para  $k \in O_s$  faça
3     para  $l = 1$  até  $n$  faça
4        $d_{ij} \leftarrow \infty$ 
5       para  $m \in O_s$  faça
6         Se  $d_{kl} > \alpha c_{km} + c_{ml}$  então
7            $d_{kl} \leftarrow \alpha c_{km} + c_{ml}$ 
8         fim
9       fim
10      fim
11     fim
12    para  $i = 1$  até  $n$  faça
13      para  $j = 1$  até  $n$  faça
14         $c'_{ij} \leftarrow \infty$ 
15        para  $k \in O_s$  faça
16          Se  $c'_{ij} > c_{ik} + d_{kj}$  então
17             $c'_{ij} \leftarrow c_{ik} + d_{kj}$ 
18          fim
19        fim
20      fim
21    fim
22 fim
  
```

---

Geralmente,  $|O_s|$  é muito menor que  $n$ , isso faz o algoritmo modificado ser muito mais eficiente para calcular o menor caminho entre todos os pontos. Esta versão precisa de  $n^2|O_s| + |O_s|^2n$  operações para obter uma solução. A tabela 8 mostra a diferença em relação a quantidade de iterações dos dois algoritmos, é possível perceber a eficiência do algoritmo modificado.

### 5.2.3 Método de construção

Os métodos de construção implementados para o problema abordado neste capítulo são baseados no método  $M1$  aplicado a rede do tipo E-R com alocação simples. Diferentemente da variante com alocação simples, no problema tratado neste capítulo o método 2 ( $M2$ ) não apresentou bons resultados. A qualidade das soluções obtidas pelas adaptações dos métodos  $M1$  e  $M2$ , na totalidade das instâncias, são equivalentes. Além disso, o

Tabela 8 – Número de operações para obter o menor caminho entre todos os pontos de um grafo.

$ O_s /n$	Floyd-Warshall	Floyd-Warshall Modificado
2/10	1000	240
5/100	$10^6$	52500
6/200	$8 * 10^6$	247200
15/1000	$10^9$	15225000

tempo para obter uma solução utilizando  $M2$  é significativamente superior a  $M1$ . Isto pode ser explicado devido a necessidade de avaliar uma nova solução de forma completa, diferentemente da situação anterior onde eram feitas avaliações parciais. Além disso, ao buscar um caminho mínimo com  $|O_s|$  assumindo grandes valores, é necessário um número significativamente superior de iterações, que resultada em um tempo maior para avaliar uma solução. Por exemplo, considere uma situação onde  $n = 100$  e ambos métodos retornam uma solução quando  $p = 5$ . Considerando somente a parte de encontrar o menor caminho entre os pontos,  $M1$  e  $M2$  demandaria, respectivamente, 155500 e 82232000 operações para gerar uma solução. Por este motivo não consideramos nesta seção a adaptação do método  $M2$  para o problema tratado neste capítulo.

O primeiro método consiste em escolher um concentrador inicial para compor uma solução  $S$ , e posteriormente melhora-la, iterativamente, inserindo concentradores adicionais, até que o custo da solução não possa ser melhorada. Considerando dois conjuntos  $O_s$  e  $C_s$  representando, respectivamente, o conjunto dos nós concentradores e não concentradores de uma solução  $S$ . Ainda, sendo  $O_s \cap C_s = \emptyset$  e  $O_s \cup C_s = N$ . O primeiro nó  $j \in C_s$  escolhido para se tornar concentrador é aquele que retorna o menor valor de função objetivo  $\phi'$ .

Posteriormente, para cada nó  $j \in C_s$ , a função objetivo é recalculada  $\phi'^{+j}$  após transformar  $j$  em concentrador, ou seja,  $C_s = C_s \setminus \{j\}$  e  $O_s = O_s \cup \{j\}$ . Caso seja encontrada uma solução melhor que a corrente  $\phi'^*$ , ou seja, uma solução onde  $\phi'^{+j} < \phi'^*$  :  $j \in C_s$ ,  $j$  é transformado em concentrador e novamente são avaliados a inserção de novos concentradores. Caso contrário, o método encerra retornando a solução corrente. Para melhor identificação deste método no restante do texto, vamos identifica-lo como  $MC^1$ .

Para instâncias de grande escala, onde  $n > 1000$ , este método pode demandar um alto tempo computacional para gerar uma solução de qualidade. Devido a este fato, com o objetivo de diminuir o tempo demandado, o conjunto  $C_s$  foi limitado a 30% do total de nós da instância. Os nós escolhidos como candidatos a concentradores foram aqueles que possuem a maior quantidade de fluxo como origem e destino. O tamanho de  $C_s$  foi definido em 30% devido ao fato de ser o menor valor com o menor impacto na função objetivo e que permitiu que as soluções fossem geradas mais rapidamente. Já a tática em utilizar a quantidade de fluxo para definir o conjunto  $C_s$ , foi definida após uma série de testes

utilizando informações como o custo de instalação, distância entre os nós e quantidade de fluxo. Este método será identificado como  $MC^2$ . Além disso, para ambos os métodos o cálculo dos diferentes  $\phi'^{+j}$  são feitos em paralelo.

## 5.2.4 Estruturas de vizinhança

Foram utilizadas três estruturas de vizinhanças pelo método de busca local para o problema tratado neste capítulo, denominadas "Troca de Função", "Remover Concentrador" e "Adicionar Concentrador". Estas estruturas são descritas a seguir:

- Troca de Função: para cada nó  $i \in C_s$  são feitas trocas com os concentradores  $k \in O_s$ . Como a troca de função entre nós que estão distantes, na maioria das vezes, não resulta em melhora na função objetivo, a troca de função entre os nós  $i$  e  $k$  foi restringida em até 2 elementos. Ou seja, para cada nó  $i \in C_s$ , são efetuadas no máximo duas trocas com os nós  $k \in O_s$ , sendo os nós  $k$  escolhidos aqueles que estão mais próximos de  $i$ . Assim, para cada troca são feitas as seguintes operações:  $C_s = C_s \setminus \{i\}$ ;  $O_s = O_s \cup \{i\}$ ;  $C_s = C_s \cup \{k\}$  e  $O_s = O_s \setminus \{k\}$ . O número de vizinhos gerados a partir dessa vizinhança é  $2|C_s|$ , e a quantidade de operações (no pior caso) para avaliar todas as soluções é aproximadamente  $2|C_s|(n^2|O_s| + |O_s|^2n + n^2)$ .
- Adicionar Concentrador: para cada nó não concentrador  $i \in C_s$ , são feitas as seguintes operações:  $C_s = C_s \setminus \{i\}$  e  $O_s = O_s \cup \{i\}$  e verificado as possíveis novas alocações dos nós pertencentes a  $C_s$ . O número de vizinhos gerados a partir dessa vizinhança é  $|C_s|$ , e a quantidade de operações (no pior caso) para avaliar todas as soluções é aproximadamente  $|C_s|(n^2|O_s| + |O_s|^2n + n^2)$ .
- Remover Concentrador: para cada nó concentrador  $k \in O_s$ , são feitas as seguintes operações:  $C_s = C_s \cup \{k\}$  e  $O_s = O_s \setminus \{k\}$  e verificado as possíveis novas alocações dos nós pertencentes a  $C_s$ . O número de vizinhos gerados a partir dessa vizinhança é  $|O_s|$ , e a quantidade de operações (no pior caso) para avaliar todas as soluções é aproximadamente  $|O_s|(n^2|O_s| + |O_s|^2n + n^2)$ .

Foi utilizado dois critérios para exploração destas estruturas de vizinhança, conhecidos como Melhor aprimorante e Primeiro de melhora. O critério Primeiro de melhora explora as soluções da estrutura de vizinhança até que uma solução melhor que a solução corrente seja encontrada. O método ILS-RVND1 utiliza este critério para explorar o espaço de soluções na busca local.

Já o Melhor aprimorante verifica toda a vizinhança, e será utilizado pelo método ILS-RVND2. Embora este tipo de exploração possa ser computacionalmente caro devido ao tamanho de certas estruturas de vizinhança, a exploração de forma paralela contribui para

um melhor desempenho do algoritmo. Na exploração da vizinhança foi utilizado o conceito de paralelização a nível de iteração, mais especificamente, a vizinhança foi particionada e explorada por diversas *threads* ao mesmo tempo.

### 5.3 Perturbação e *Path-Relinking*

O procedimento de perturbação altera a estrutura de uma solução utilizando as estruturas de vizinhança "Adicionar e Remover Concentrador". Ao todo, são alterados  $\pi n$  posições de uma solução, onde  $\pi$  determina o nível da perturbação. Estas perturbações necessitam ser forte o suficiente para permitir que o método de busca local explore diferentes soluções e escape de ótimos locais. Ao mesmo tempo, deve ser suficientemente fraca para não deslocar a pesquisa para posições muito distante, isto poderia levar a uma busca aleatória. Não menos importante, vale lembrar que o procedimento de perturbação é aplicado de acordo com a tática explicada no capítulo 2.

Já a ideia do método *Path-Relinking* utilizado neste problema é a mesma utilizada no problema com alocação simples (ver Capítulo 4 seção 4.2.5). Sendo o método "Realocação" utilizado no PR substituído pela versão modificada do Floyd-Warshall.

## 5.4 Experimentos Computacionais

Esta seção apresenta os resultados obtidos pelas diferentes propostas do método ILS-RVND proposto neste trabalho. Neste experimento foi utilizado um computador com dois processadores modelo Intel Xeon E5-2620 de 2.1 Ghz, cada um contendo 6 núcleos, totalizando 24 *threads*(M3). Além disso, a maquina possui 64Gb RAM e executa como sistema operacional um sistema baseado em Unix (Linux).

### 5.4.1 Métodos de Construção

Nesta seção são apresentados os resultados obtidos pelos métodos de construção propostos. As tabelas 9 e 10 mostram os valores de função objetivo para 4 versões destes métodos, sendo elas as versões sequências e paralelas de  $MC^1$  e  $MC^2$ , sinalizadas com  $p$  para o método paralelo e  $s$  sequencial. Os métodos foram limitados à  $n$  segundos de processamento.

Para instâncias de até 1000 nós, a qualidade das soluções obtidas pelas versões sequenciais e paralelas, de cada método, é a mesma. Isto porque o método sequencial é capaz de gerar a solução dentro do tempo limite definido neste trabalho. Para instâncias onde  $n > 1000$  isto não se repete, neste caso o método sequencial não obteve uma solução dentro do tempo limite, e os valores apresentados consistem nas soluções parciais obtidas por estes métodos. Os tempos são apresentados na tabela 11, em média, a obtenção de

uma solução pelas versões paralelas dos métodos é 5 vezes mais rápida. A Figura 35 mostra o *speedup* obtido pelos métodos paralelos agrupados pelo tamanho da instância, onde as linhas tracejada e contínua representam, respectivamente, os métodos  $MC^2$  e  $MC^1$ .

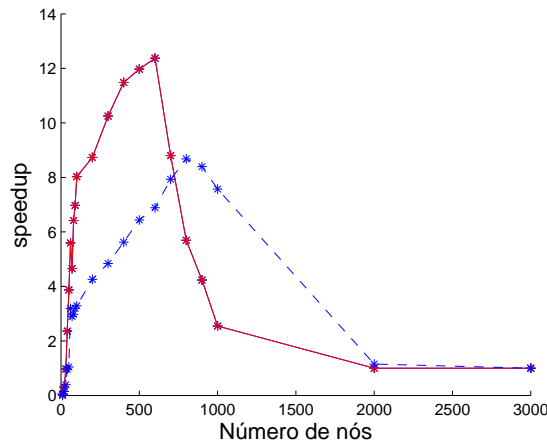


Figura 34 – *speedup* calculado para o método de construção paralelo em relação ao sequencial

O *speedup* para instâncias onde  $n > 1000$  é 1 devido ao fato de ambas as versões, sequencial e paralela, demandarem o limite de tempo computacional para gerar uma solução. Além disso, o ganho de tempo ao paralelizar  $MC^2$  é menor em relação a  $MC^1$ , isso pode ser explicado pela quantidade de operações necessárias serem significativamente inferiores, e a relação tempo de processamento  $\times$  sobrecarga impactar no ganho de *speedup*. Os outros motivos em relação a perda de desempenho de *speedup* consistem nos mesmo apresentados no Capítulo anterior (ver seção 4.3.1).

Ainda, considerando a qualidade das soluções obtidas por estes métodos, é possível perceber que  $MC^{2p}$  obtém soluções inferiores, de pior qualidade, em relação a  $MC^{1p}$  na maioria das instâncias. Por outro lado, o tempo para obter soluções utilizando  $MC^2$  chega a ser 3,5 vezes inferior a  $MC^1$ . As Figuras 35 e 36 representam, respectivamente, o *gap* entre as soluções geradas por  $MC^1$  e  $MC^2$  e o *speedup* de  $MC^2$  em relação a  $MC^1$ , ambos em relação aos números de nós.

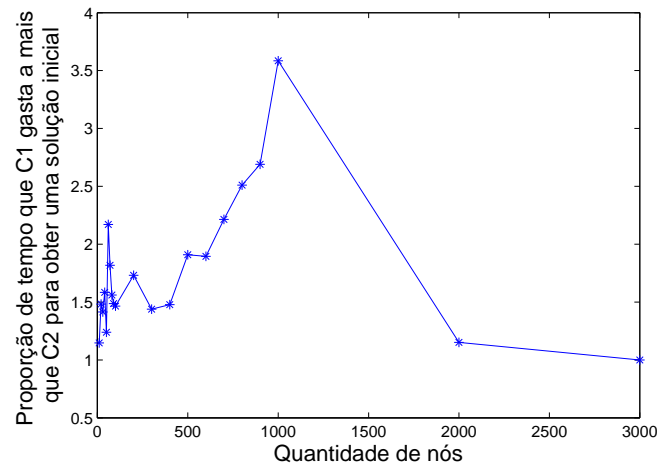


Figura 35 – Proporção de tempo gasto por  $C1$  quando comparado a  $C2$  para obter uma solução em relação ao número de nós

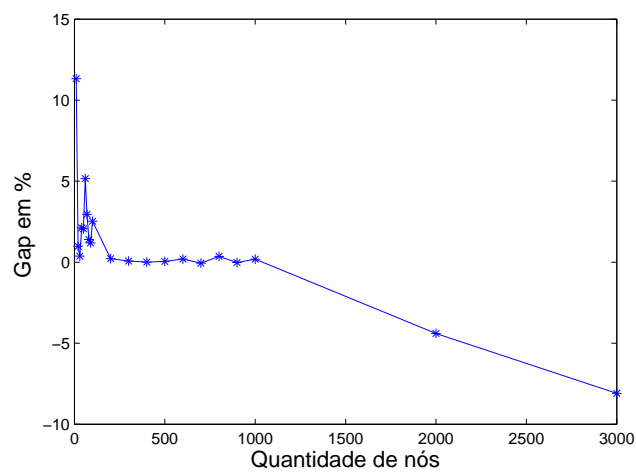


Figura 36 – Gap em relação a função objetivo entre  $C2$  e  $C1$

Tabela 9 – Valor da função objetivo obtida pelas variações dos métodos de construção propostos.

Instância	$C1^p$	$C2^p$	$C1^s$	$C2^s$	Instância	$C1^p$	$C2^p$	$C1^s$	$C2^s$
HL10_2	30704,98	37164,85	30704,98	37164,85	HL80_2	2680073,58	2727803,79	2680073,58	2727803,79
HL10_4	33741,05	38256,27	33741,05	38256,27	HL80_4	2959944,25	3016302,82	2959944,25	3016302,82
HL10_6	35976,22	39200,81	35976,22	39200,81	HL80_6	3155223,52	3197152,97	3155223,52	3197152,97
HL10_8	37927,01	39554,51	37927,01	39554,51	HL80_8	3281134,16	3306821,28	3281134,16	3306821,28
HL20_2	186654,70	189412,74	186654,70	189412,74	HL90_2	3487178,65	3567529,56	3487178,65	3567529,56
HL20_4	206761,79	208165,53	206761,79	208165,53	HL90_4	3887370,76	3936825,57	3887370,76	3936825,57
HL20_6	218256,82	221577,03	218256,82	221577,03	HL90_6	4140351,99	4151678,27	4140351,99	4151678,27
HL20_8	225927,75	226800,69	225927,75	226800,69	HL90_8	4249335,63	4298122,39	4249335,63	4298122,39
HL30_2	341581,07	355331,28	341581,07	355331,28	HL100_2	3706289,21	3900549,55	3706289,21	3900549,55
HL30_4	390771,72	390295,82	390771,72	390295,82	HL100_4	4227195,44	4321872,27	4227195,44	4321872,27
HL30_6	418018,27	416454,61	418018,27	416454,61	HL100_6	4577987,73	4694741,16	4577987,73	4694741,16
HL30_8	437291,37	431605,17	437291,37	431605,17	HL100_8	4779318,20	4822452,58	4779318,20	4822452,58
HL40_2	615517,85	634599,86	615517,85	634599,86	HL200_2	16574411,96	16781185,39	16574411,96	16781185,39
HL40_4	684900,50	709657,75	684900,50	709657,75	HL200_4	18051623,68	18151931,04	18051623,68	18151931,04
HL40_6	754171,43	770326,31	754171,43	770326,31	HL200_6	18964266,37	18935630,20	18964266,37	18935630,20
HL40_8	805186,81	807214,98	805186,81	807214,98	HL200_8	19422742,68	19306542,96	19422742,68	19306542,96
HL50_2	816019,74	853367,69	816019,74	853367,69	HL300_2	33242023,53	33300008,45	33242023,53	33300008,45
HL50_4	923674,97	942052,86	923674,97	942052,86	HL300_4	36997895,66	36875193,18	36997895,66	36875193,18
HL50_6	1011081,50	1027953,48	1011081,50	1027953,48	HL300_6	39518362,43	39581916,75	39518362,43	39581916,75
HL50_8	1068244,45	1076171,46	1068244,45	1076171,46	HL300_8	41181286,64	41286398,35	41181286,64	41286398,35
HL60_2	1279075,01	1407509,50	1279075,01	1407509,50	HL400_2	59409158,15	59377720,81	59409158,15	59377720,81
HL60_4	1482429,20	1576425,29	1482429,20	1576425,29	HL400_4	66750516,48	66904824,63	66750516,48	66904824,63
HL60_6	1644394,93	1707377,82	1644394,93	1707377,82	HL400_6	72131274,53	71917669,15	72131274,53	71917669,15
HL60_8	1732048,31	1780354,31	1732048,31	1780354,31	HL400_8	75225376,78	75341305,38	75225376,78	75341305,38
HL70_2	1895100,23	1984361,06	1895100,23	1984361,06	HL500_2	83614654,43	83374124,47	83614654,43	83374124,47
HL70_4	2128856,25	2197392,15	2128856,25	2197392,15	HL500_4	94870216,34	94955213,29	94870216,34	94955213,29
HL70_6	2291059,01	2349671,85	2291059,01	2349671,85	HL500_6	103137907,71	103205374,82	103137907,71	103205374,82
HL70_8	2376992,85	2425207,67	2376992,85	2425207,67	HL500_8	108483577,90	108744755,43	108483577,90	108744755,43

Tabela 10 – Valor da função objetivo obtida pelas variações dos métodos de construção propostos.

Instância	$C1^p$	$C2^p$	$C1^s$	$C2^s$
HL600_2	117344790,46	118329117,18	117344790,46	118329117,18
HL600_4	134554529,30	134802090,89	134554529,30	134802090,89
HL600_6	147502985,64	147035065,27	147502985,64	147035065,27
HL600_8	155794222,58	156096589,22	155794222,58	156096589,22
HL700_2	154010634,78	153612538,02	155916048,56	153612538,02
HL700_4	177049014,94	176779728,71	177355825,91	176779728,71
HL700_6	194551516,59	194654494,02	194551516,59	194654494,02
HL700_8	205216738,52	205328062,24	205216738,52	205328062,24
HL800_2	187797500,10	188655551,76	199759792,00	188655551,76
HL800_4	219090226,09	219594384,29	224486893,80	219594384,29
HL800_6	242599675,50	243791591,94	244269770,14	243791591,94
HL800_8	258072433,41	258828062,31	258614804,33	258828062,31
HL900_2	238545397,76	238128139,20	262995623,34	238354294,17
HL900_4	279206050,34	279195757,09	292689921,45	279195757,09
HL900_6	310186963,29	310390038,25	316494679,51	310390038,25
HL900_8	330087841,73	330063151,05	333592346,68	330063151,05
HL1000_2	287126343,30	286529186,83	341639299,48	292080470,51
HL1000_4	337769981,36	337820425,88	362559390,44	339582599,76
HL1000_6	376978240,22	379081184,66	398860545,89	379081184,66
HL1000_8	403947383,36	405002971,63	415646908,33	405002971,63
HL2000_2	1104727276,76	994235830,90	1804444005,80	1330163856,90
HL2000_4	1287164725,48	1212644493,95	1843586816,09	1470388684,93
HL2000_6	1434935467,48	1397981055,76	1867049043,76	1565530143,66
HL2000_8	1543678724,80	1539410016,51	1879040981,92	1624888984,50
HL3000_2	2652658400,19	2297986063,26	4534408100,44	3615403247,48
HL3000_4	3015708742,00	2733728200,60	4534408100,44	3798417875,11
HL3000_6	3297039832,45	3107582192,38	4534408100,44	3920031568,91
HL3000_8	3499367251,01	3392125649,65	4534408100,44	3981211838,07

Tabela 11 – Tempo gasto em segundos para obter uma solução pelas variações dos métodos de construção propostos.

Instância	$C1^P$	$C2^P$	$C1^s$	$C2^s$	Instância	$C1^P$	$C2^P$	$C1^s$	$C2^s$	Instância	$C1^P$	$C2^P$	$C1^s$	$C2^s$
HL10_2	0,03	0,024	0,00	0,00	HL80_2	0,11	0,08	0,70	0,22	HL600_2	47,71	24,59	580,51	172,35
HL10_4	0,03	0,02	0,00	0,00	HL80_4	0,09	0,05	0,50	0,15	HL600_4	48,31	23,68	600,09	176,13
HL10_6	0,03	0,01	0,00	0,00	HL80_6	0,06	0,07	0,59	0,15	HL600_6	28,42	16,04	354,34	102,25
HL10_8	0,00	0,02	0,00	0,00	HL80_8	0,10	0,02	0,44	0,14	HL600_8	27,94	16,10	350,69	103,29
HL20_2	0,04	0,04	0,02	0,00	HL90_2	0,14	0,09	0,91	0,28	HL700_2	106,57	52,08	700,18	425,55
HL20_4	0,03	0,01	0,01	0,01	HL90_4	0,13	0,07	0,92	0,23	HL700_4	79,28	40,58	700,13	324,79
HL20_6	0,03	0,01	0,01	0,00	HL90_6	0,09	0,07	0,53	0,22	HL700_6	67,57	20,78	700,05	160,50
HL20_8	0,03	0,02	0,01	0,00	HL90_8	0,06	0,05	0,53	0,16	HL700_8	46,85	22,14	542,50	164,16
HL30_2	0,04	0,02	0,05	0,01	HL100_2	0,25	0,14	2,01	0,46	HL800_2	235,19	86,34	800,15	800,07
HL30_4	0,04	0,04	0,04	0,01	HL100_4	0,15	0,10	1,26	0,37	HL800_4	139,35	57,89	800,13	487,84
HL30_6	0,04	0,01	0,04	0,01	HL100_6	0,13	0,11	1,02	0,29	HL800_6	102,09	43,20	800,11	357,38
HL30_8	0,01	0,03	0,02	0,01	HL100_8	0,09	0,07	0,73	0,28	HL800_8	85,63	36,55	800,03	298,84
HL40_2	0,04	0,05	0,13	0,03	HL200_2	1,00	0,49	7,62	2,32	HL900_2	357,58	120,72	900,21	900,17
HL40_4	0,06	0,05	0,13	0,03	HL200_4	0,74	0,50	7,66	2,31	HL900_4	252,90	86,66	900,16	803,90
HL40_6	0,05	0,01	0,09	0,03	HL200_6	0,68	0,34	5,71	1,22	HL900_6	144,83	66,45	900,05	601,46
HL40_8	0,05	0,01	0,11	0,02	HL200_8	0,45	0,33	4,05	1,21	HL900_8	95,25	42,35	900,10	347,93
HL50_2	0,05	0,06	0,30	0,06	HL300_2	3,11	2,41	36,34	12,43	HL1000_2	622,92	188,50	1000,02	1000,32
HL50_4	0,06	0,05	0,20	0,04	HL300_4	2,66	2,10	26,85	10,01	HL1000_4	508,87	137,46	1000,05	1000,09
HL50_6	0,05	0,02	0,16	0,05	HL300_6	2,93	1,84	26,42	7,77	HL1000_6	273,64	71,05	1000,08	834,82
HL50_8	0,06	0,04	0,16	0,03	HL300_8	2,61	1,51	26,21	7,84	HL1000_8	164,69	41,04	1000,10	482,32
HL60_2	0,10	0,03	0,57	0,13	HL400_2	10,38	7,46	120,94	43,90	HL2000_2	2000,15	2000,52	2000,11	2000,55
HL60_4	0,08	0,03	0,42	0,13	HL400_4	8,83	6,02	99,65	36,90	HL2000_4	2000,00	2000,47	2000,38	2000,53
HL60_6	0,08	0,06	0,53	0,10	HL400_6	9,24	4,74	106,52	23,97	HL2000_6	2000,02	1974,55	2000,25	2000,67
HL60_8	0,04	0,02	0,20	0,08	HL400_8	5,46	4,72	62,23	24,25	HL2000_8	2000,15	969,85	2000,60	2000,67
HL70_2	0,09	0,05	0,43	0,16	HL500_2	32,84	17,07	395,93	117,19	HL3000_2	3000,71	3000,20	3000,15	3000,03
HL70_4	0,08	0,05	0,33	0,10	HL500_4	28,12	13,16	340,37	85,74	HL3000_4	3000,04	3000,48	3000,69	3001,15
HL70_6	0,08	0,04	0,37	0,12	HL500_6	17,19	9,30	196,22	58,91	HL3000_6	3000,09	3000,39	3000,85	3000,47
HL70_8	0,04	0,02	0,22	0,09	HL500_8	13,43	8,46	163,92	46,99	HL3000_8	3000,11	3000,33	3000,70	3000,64

É possível observar que para a maioria das instâncias o *gap* está próximo de 1%, além disso, onde os valores são negativos indicam que  $MC^2$  obteve melhores soluções. Assim, considerando a relação de tempo × qualidade de solução, optamos em utilizar o método  $MC^2$ , que além de ser mais rápido, apresenta melhores soluções para os problemas de maior dimensão. Baseados nos resultados, foi concluído que elencar os possíveis candidatos a concentradores de acordo com os fluxos de entrada e saída foi uma boa tática.

## 5.5 Comparação entre ILS-RVND1 × ILS-RVND2

Nesta seção serão comparados o desempenho entre as duas variantes da meta-heurística ILS-RVND-PR propostas neste trabalho, denominados ILS-RVND1 e ILS-RVND2. Os algoritmos foram configurados de acordo com as informações do capítulo anterior, ou seja, o número de *threads* e o grau de perturbação  $\pi$  foram mantidos os mesmos. Inicialmente, foram computados os seguintes indicadores: *KBS*, DevMin e DevMed. Como critério de parada para os algoritmos foi utilizado o tempo de máquina, limitado a  $n$  segundos. Todos os indicadores foram contabilizados após 10 execuções de cada método, considerando as 84 instâncias. Os valores obtidos são mostrados na tabela 12.

Tabela 12 – Valores obtidos para alguns indicadores referentes aos métodos ILS-RVND1 e ILS-RVND2

Indicadores	ILS-RVND1	ILS-RVND2
KBS	817	793
DevMin	3,92667E-05	6,59833E-05
DevMed	3,93527E-05	6,60891E-05

Novamente, o comportamento dos algoritmos foram parecidos. Porém, para estes indicadores, ao contrário da variante com alocação simples, quem se destacou foi o método ILS-RVND1. Considerando as 840 replicações, este método encontrou soluções melhores ou de mesma qualidade, em relação a ILS-RVND2, em aproximadamente 95% das replicações. Os valores de DevMin e DevMed mostram que os algoritmos convergem para as mesma soluções, mesmo utilizando sementes aleatórias. A tabela 13 mostra os valores médios de função objetivo obtidos neste experimento, sendo os valores diferenciados de função objetivo destacados em negrito, nestes casos, um dos algoritmos foi melhor em relação ao outro.

Para verificar a convergência destes métodos em relação ao tempo de execução, foram feitos alguns gráficos TTT. Desta vez as instâncias foram escolhidas estrategicamente, o objetivo foi observar o comportamento destes algoritmos nas instâncias onde apresentam diferença na função objetivo. A linha contínua representa o método ILS-RVND2, enquanto a pontilhada o método ILS-RVND1. Como alvo foi definida a função objetivo obtida

pelo algoritmo com pior desempenho. Os resultados são apresentados na Figura 37. Para instância HL500\_4 não foi apresentado o gráfico devido ao fato da solução inicial ser igual a pior solução obtida entre os algoritmos, neste caso, as curvas seriam as mesmas. Considerando as instâncias onde HL200\_2 e HL300\_4, é possível perceber que o método ILS-RNVD2 alcança uma solução de qualidade mais rapidamente, porém, ao verificar as soluções finais obtidas pelos métodos é perceptível que este fica preso em um ótimo local.

Esta diferença de comportamento pode ser explicada pelo número inferior de soluções exploradas paralelamente. Isto porque ao diminuir a quantidade de *threads* de busca do método ILS-RVND1 houve perda de desempenho. Um número menor de *threads* de busca diminui a segmentação do espaço para aplicação da perturbação, bem como o número de soluções disponíveis para o processo de intensificação PR. Estes fatos também pode ter colaborado para perde de desempenho.

Para instância HL400\_6, onde ILS-RVND2 obteve um melhor desempenho, verificamos que, em média, o tempo para obter a solução alvo por ILS-RVND1 está próximo do tempo limite de processamento, assim, seria necessário mais tempo computacional para que a solução obtida por este algoritmo fosse melhorada. Ao aumentar o tempo de processamento para  $1,2n$  o método ILS-RVND1 alcançou o mesmo resultado. Comportamento que não se repete para instância HL500\_4.

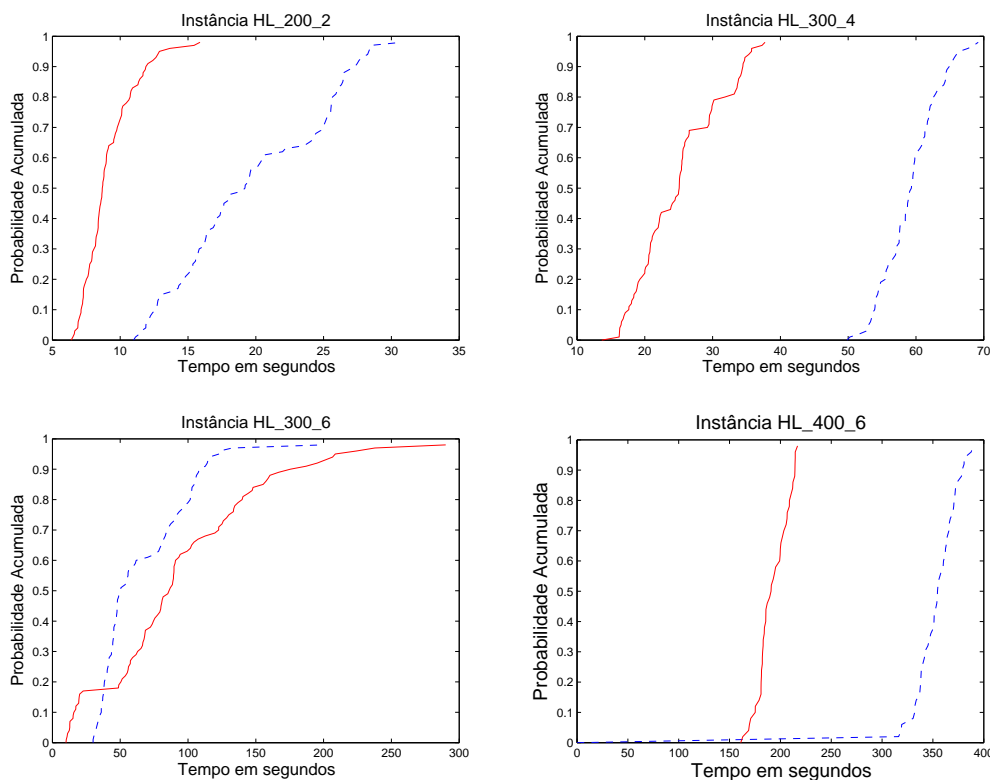


Figura 37 – Gráficos TTT.

Levando em consideração o desempenho dos métodos, em relação a qualidade

das soluções obtidas e tempo de convergência, foi escolhido o ILS-RVND1 para avaliar seu comportamento em relação a outros métodos da literatura. Esta escolha considerou o tempo disponibilizado para os experimentos ( $n$  segundos para cada instância). Em situações onde o tempo de processamento disponibilizado é menor, o método ILS-RVND2 poderia ser considerado mais eficiente. Este comportamento pode ser observado no capítulo anterior, onde foi disponibilizado apenas  $0.4n$  segundo de processamento, naquela ocasião o método que se destacou foi o ILS-RVND2.

Tabela 13 – Valores médios de função objetivo obtidos pelas variações dos métodos propostos neste trabalho (ILS-RVND1 e ILSRVND2).

Instância	ILS-RVND1	ILS-RVND2	Instância	ILS-RVND1	ILS-RVND2	Instância	ILS-RVND1	ILS-RVND2
HL10_2	30493,55	30493,5547	HL80_2	2585300,12	2585300,12	HL600_2	118055305,79	118055305,79
HL10_4	33657,81	33657,81	HL80_4	2891689,89	2891689,89	HL600_4	134802090,89	134802090,89
HL10_6	35976,22	35976,22	HL80_6	3101536,31	3101536,31	HL600_6	147035065,27	147035065,27
HL10_8	37927,01	37927,01	HL80_8	3195750,88	3195750,88	HL600_8	156096589,22	156096589,22
HL20_2	180948,12	180948,12	HL90_2	3389391,92	3389391,92	HL700_2	153279802,38	153279802,38
HL20_4	200803,40	200803,40	HL90_4	3775711,31	3775711,31	HL700_4	176779728,71	176779728,71
HL20_6	214225,78	214225,78	HL90_6	4008828,77	4008828,77	HL700_6	194654494,02	194654494,02
HL20_8	219218,08	219218,08	HL90_8	4123570,07	4123570,07	HL700_8	205328062,24	205328062,24
HL30_2	329474,51	329474,51	HL100_2	3649881,27	3649881,27	HL800_2	188655551,76	188655551,76
HL30_4	378048,96	378048,96	HL100_4	4153455,83	4153455,83	HL800_4	219594384,29	219594384,29
HL30_6	410102,11	410102,11	HL100_6	4498936,84	4498936,84	HL800_6	243791591,94	243791591,94
HL30_8	429472,92	429472,92	HL100_8	4694680,70	4694680,70	HL800_8	258828062,31	258828062,31
HL40_2	589357,92	589357,92	HL200_2	<b>16256922,23</b>	16333821,05	HL900_2	238128139,20	238128139,20
HL40_4	671659,53	671659,53	HL200_4	17557819,13	17557819,13	HL900_4	279187568,27	279187568,27
HL40_6	739614,71	739614,71	HL200_6	18505441,12	18505441,12	HL900_6	310390038,25	310390038,25
HL40_8	780267,78	780267,78	HL200_8	18931200,99	18931200,99	HL900_8	330063151,05	330063151,05
HL50_2	803690,59	803690,59	HL300_2	32762267,76	32762267,76	HL1000_2	286529186,83	286529186,83
HL50_4	910278,02	910278,02	HL300_4	<b>36756973,31</b>	36761133,15	HL1000_4	337820425,88	337820425,88
HL50_6	986908,47	986908,47	HL300_6	<b>39307636,74</b>	39320941,67	HL1000_6	379081184,66	379081184,66
HL50_8	1034579,41	1034579,41	HL300_8	40713818,08	40713818,08	HL1000_8	405002971,63	405002971,63
HL60_2	1267175,56	1267175,56	HL400_2	57688992,20	57688992,20	HL2000_2	994235830,90	994235830,90
HL60_4	1467762,91	1467762,91	HL400_4	65555593,14	65555593,14	HL2000_4	1212644493,95	1212644493,95
HL60_6	1624265,42	1624265,42	HL400_6	71241311,32	<b>71223844,78</b>	HL2000_6	1397981055,76	1397981055,76
HL60_8	1715354,54	1715354,54	HL400_8	74411122,51	74505896,28	HL2000_8	1539410016,51	1539410016,51
HL70_2	1881663,45	1881663,45	HL500_2	83374124,47	83374124,47	HL3000_2	2297986063,26	2297986063,26
HL70_4	2123045,60	2123045,60	HL500_4	94955213,29	<b>94666181,79</b>	HL3000_4	2733728200,60	2733728200,60
HL70_6	2284389,61	2284389,61	HL500_6	103205374,82	103205374,82	HL3000_6	3107582192,38	3107582192,38
HL70_8	2355127,23	2355127,23	HL500_8	<b>107217073,06</b>	107426447,51	HL3000_8	3392125649,65	3392125649,65

## 5.6 Comparação com a Literatura

Nesta seção será apresentada a comparação da metodologia proposta, ILS-RVND1, em relação a alguns trabalhos da literatura. Foram escolhidos três diferentes metodologias, um algoritmo genético (KRATICA et al., 2007), um algoritmo híbrido composto pelas meta-heurísticas Busca tabu e *Simulated Annealing* (CHEN, 2007) e um algoritmo Memético (DJUKIC, 2010). A avaliação de desempenho entre os algoritmos estudados foi realizado por meio de um teste estatístico, que visa identificar se existe diferença estatística significativa entre eles. Esta análise foi feita utilizando dados obtidos por cada algoritmo, contendo o valor da função objetivo após 10 execuções independentes. Como critério de parada foi estabelecido o tempo máximo de  $n$  segundos, ou seja, o número de nós da instância. Os testes foram executados de maneira aleatorizada, tanto em relação a instância quanto ao algoritmo. Além disso, os algoritmos utilizaram a mesma semente geradora de números aleatórios em cada replicação de uma determinada instância. Tais procedimentos garantem condições experimentais homogêneas e independência dos erros, evitando a introdução de efeitos espúrios e tendências nas amostras (MONTGOMERY; RUNGER, 2003).

Além disso, alguns indicadores foram utilizados para aferir o desempenho dos mesmo. A tabela 14 mostra os indicadores KBS, DevMin e DevMed obtidos no experimento.

Tabela 14 – Valores obtidos para alguns indicadores para métodos da literatura e o algoritmo proposto neste trabalho

Indicadores	ILS-RVND1	Memético	Genético	<i>Simulated Annealing</i>
KBS	756	452	271	151
DevMin	0,0003	0,0099	0,0181	0,0959
DevMed	0,0004	0,0108	0,0182	0,0989

Novamente, os valores obtidos para estes indicadores mostram a eficiência do método proposto. O algoritmo apresentou resultados, melhores ou iguais aos outros algoritmos, em aproximadamente 90% das instâncias. Além disso, os valores de DevMin e DevMed mostram que a variabilidade das soluções obtidas é pequena, mostrando a boa convergência deste algoritmo. Os resultados também confirmam a melhoria no algoritmo proposto por Kratica et al. (2007) feita no trabalho apresentado por Djukic (2010). A inserção da busca local melhorou significativamente as soluções obtidas.

Já o algoritmo apresentado por Chen (2007) não apresentou bons resultados. Foi observado que para instâncias onde soluções de qualidade possuem vários concentradores, este algoritmo não possui bom desempenho. Isso pode ser explicado devido ao fato deste algoritmo resolver vários problemas  $p$ -UMLHLP, e para cada problema o tempo demandado para obter uma solução, de acordo com os critérios utilizados, pode ser consideravelmente alto. Por exemplo, para uma instância onde  $n = 200$  e a combinação de concentradores

para obter uma solução ótima seja igual 10, seria necessário aplicar a heurística para 10 diferentes problemas. Quanto maior o valor de  $p$ , mais tempo é demandado para convergência do algoritmo aplicado ao  $p$ -UMLHLP, assim, em muitas das instâncias o tempo disponibilizado não foi suficiente para convergência.

As tabelas 15-16 mostram os valores médios de função objetivo obtidos por cada algoritmo após 10 execuções. Além disso, a última coluna da tabela 15 mostra os valores de função objetivo obtidos pelo método de Benders (CAMARGO; Miranda Jr; LUNA, 2008), limitado a um dia de processamento. É possível observar que para todas as instâncias onde foi possível obter o valor ótimo, ou seja, o valor do *gap* igual a 0, o algoritmo proposto neste trabalho alcançou os mesmos resultados. Para as outras, o algoritmo proposto obteve melhores soluções. Vale ressaltar que embora os métodos heurísticos não possam garantir otimizabilidade, de forma geral, eles tem se mostrado eficientes para alcançar soluções de qualidade em um tempo consideravelmente reduzido, como pode ser visto neste trabalho. O método de Benders foi aplicado somente a instâncias onde  $n \leq 200$  por demandar tempo significativamente superior para instâncias maiores.

Os indicadores apresentados até o momento mostram a superioridade do ILS-RVND1. Isso foi confirmado ao se aplicar o teste estatístico e o procedimento de múltiplas comparações. Considerando o nível de significância igual a 0,05, obtivemos o valor de  $p = 2.1873e-31$ , o que descarta a hipótese que todos os algoritmos são estatisticamente iguais. Ao aplicar o procedimento de múltiplas comparações, verificamos que o método proposto não apenas é significativamente diferente, mas também é o melhor. O gráfico obtido ao aplicar este procedimento é mostrado na Figura 38, considere os algoritmos 1,2,3 e 4 sendo, respectivamente, Algoritmo Genético, *Simulated Annealing*, Algoritmo Memético e o ILS-RVND. Observa-se que os algoritmos que apresentam melhores resultados estão a esquerda do gráfico, e os que apresentam diferença significativa os intervalos não se sobrepõem.

As Figuras 39 e 40 mostram os gráficos TTT obtidos pelos algoritmos ILS-RVND, Genético (AG) e Memético(M) para algumas instâncias, selecionadas ao acaso, com  $n$  variando de 10 a 3000. Considere a linha com o marcador '\*' sendo algoritmo proposto neste trabalho, a linha tracejada o Algoritmo Genético, e a linha com o marcador triangular sendo o Algoritmo Memético. Os marcadores foram plotados de quatro a quatro pontos para melhorar a visualização do gráfico. As Figuras mostram que a probabilidade do ILS-RVND alcançar as soluções alvo, mais rapidamente, é maior. Assim, considerando todo experimento, é possível concluir que além convergir para as melhores soluções, o algoritmo proposto faz isso de forma rápida. Também é possível verificar que a inserção da busca local no algoritmo proposto por Kratica et al. (2007) melhora significativamente a probabilidade de encontrar soluções de qualidade, demandando menos tempo computacional.

Ainda, sobre o comportamento dos algoritmos, considerando os gráficos TTT, é

Tabela 15 – Valores de função objetivo obtidos pelo métodos da literatura e o algoritmo proposto neste trabalho.

Instância	Genético	SA	Memético	ILS-RVND1	Benders	%gap
HL10_2	30493,55	30493,55	30493,55	30493,55	30493,55	0
HL10_4	33657,81	33741,05	33657,81	33657,81	33657,81	0
HL10_6	35976,22	35976,22	35976,22	35976,22	35976,22	0
HL10_8	37927,01	38015,05	37927,01	37927,01	37927,01	0
HL20_2	180948,12	180948,12	180948,12	180948,12	180948,12	0
HL20_4	200803,40	200803,40	200803,40	200803,40	200803,40	0
HL20_6	214225,78	214225,78	214225,78	214225,78	214225,78	0
HL20_8	219218,08	219218,08	219218,08	219218,08	219218,08	0
HL30_2	329474,51	329474,51	329474,51	329474,51	329474,51	0
HL30_4	378048,96	378048,96	378048,96	378048,96	378048,96	0
HL30_6	410102,11	410102,11	410102,11	410102,11	410102,11	0
HL30_8	429472,92	429472,92	429472,92	429472,92	429472,92	0
HL40_2	593488,08	590806,41	589357,92	589357,92	589357,92	0
HL40_4	678073,38	679654,53	671659,53	671659,53	671659,53	0
HL40_6	739614,71	742949,59	739614,71	739614,71	739614,71	0
HL40_8	781302,26	783076,64	780267,78	780267,78	780267,78	0
HL50_2	805277,52	806628,31	803690,59	803690,59	803716,27	0,4
HL50_4	911386,08	910278,02	911386,08	910278,02	910278,02	0
HL50_6	986908,47	986908,47	986908,47	986908,47	986908,47	0
HL50_8	1034579,41	1034579,41	1034579,41	1034579,41	1034579,41	0
HL60_2	1287468,26	1302885,80	1267175,56	1267175,56	1269945,84	1,9
HL60_4	1467762,91	1488163,60	1470384,99	1467762,91	1472126,72	1,3
HL60_6	1624265,42	1633279,48	1624265,42	1624265,42	1625286,76	0,3
HL60_8	1715354,54	1721286,04	1715354,54	1715354,54	1715354,54	0
HL70_2	1888286,67	1923324,74	1881663,45	1881663,45	1881663,45	0,5
HL70_4	2123045,60	2128823,65	2123045,60	2123045,60	2123045,59	0,4
HL70_6	2284389,61	2288028,95	2284389,61	2284389,61	2284389,61	0
HL70_8	2356566,91	2355127,23	2355127,23	2355127,23	2355127,23	0
HL80_2	2590331,57	2605792,30	2585300,12	2585300,12	2600040,59	2,5
HL80_4	2893035,97	2911936,15	2891689,89	2891689,89	2893035,96	1,3
HL80_6	3104941,41	3105725,76	3101536,31	3101536,31	3102284,71	0,6
HL80_8	3195750,88	3195920,54	3195750,88	3195750,88	3195750,88	0
HL90_2	3389391,92	3449769,86	3397895,99	3389391,92	3404020,57	3,1
HL90_4	3784144,89	3789702,78	3785955,46	3775711,31	3786628,38	2,2
HL90_6	4028627,28	4010578,27	4010578,27	4008828,77	4008828,77	0,4
HL90_8	4123570,07	4123570,07	4123570,07	4123570,07	4123570,07	0
HL100_2	3667266,09	3714139,54	3649881,27	3649881,27	3659709,85	2,4
HL100_4	4158817,34	4170626,04	4153455,83	4153455,83	4163025,32	1,9
HL100_6	4503695,29	4509495,47	4498936,84	4498936,84	4508976,82	0,9
HL100_8	4697187,98	4700433,69	4694680,70	4694680,70	4694680,70	0
HL200_2	16298986,41	16315953,76	16236194,95	16236194,95	16292039,76	5,9
HL200_4	17557819,13	17647125,60	17557819,13	17557819,13	17593597,33	3,6
HL200_6	18545951,11	18511600,94	18507539,24	18505441,12	18542967,28	2,7
HL200_8	18969956,53	18969956,53	18931200,99	18931200,99	18962219,55	1,38

Tabela 16 – Valores de função objetivo obtidos pelo métodos da literatura e o algoritmo proposto neste trabalho.

Instância	Genético	SA	Memético	ILS-RVND1
HL300_2	33330797,59	34349222,12	33055716,29	32762267,76
HL300_4	37011526,41	37323537,81	36861284,00	36756973,31
HL300_6	39427264,90	39364525,08	39320941,67	39307636,74
HL300_8	40771789,17	40754517,20	40757199,15	40713818,08
HL400_2	59190685,17	66128136,82	59860820,05	57688992,20
HL400_4	66267865,39	70465710,76	66453338,53	65555593,14
HL400_6	72031436,48	73540635,27	71571102,38	71241311,32
HL400_8	74653505,13	75231247,12	74411122,51	74411122,51
HL500_2	83518905,33	97013759,51	87931376,27	83374124,47
HL500_4	94314327,98	103822205,23	95123821,61	94955213,29
HL500_6	102825661,39	107242366,76	102898788,67	103205374,82
HL500_8	107495673,52	109678864,40	107980112,79	107217073,06
HL600_2	118879703,09	141563975,50	118657394,17	118055305,79
HL600_4	135544818,97	150737855,88	134286435,44	134802090,89
HL600_6	147981187,21	156944681,61	148948881,00	147035065,27
HL600_8	155721005,77	160628879,62	156745866,11	156096589,22
HL700_2	156784001,54	188057524,89	156396783,74	153279802,38
HL700_4	178327833,73	199127180,17	177884136,74	176779728,71
HL700_6	195156983,29	207570076,16	193764629,04	194654494,02
HL700_8	206427726,65	212582506,89	204528329,20	205328062,24
HL800_2	192741389,92	269811042,28	194832193,61	188655551,76
HL800_4	220772827,02	278732460,34	224445090,30	219594384,29
HL800_6	244635518,82	284137484,05	254896129,80	243791591,94
HL800_8	259225420,47	286030573,54	257919950,86	258828062,31
HL900_2	247834029,44	350976973,23	257409344,32	238128139,20
HL900_4	281262287,52	362473045,67	281343897,66	279187568,27
HL900_6	312301300,48	369247208,77	314231629,71	310390038,25
HL900_8	333824869,59	372598392,15	331373757,77	330063151,05
HL1000_2	297559653,42	429505964,78	306064914,17	286529186,83
HL1000_4	342829810,41	443804725,55	335774087,36	337820425,88
HL1000_6	383262482,66	452287363,65	382299818,39	379081184,66
HL1000_8	407193094,87	456593058,27	404404404,42	405002971,63
HL2000_2	1114110476,23	1718990429,31	1101241950,39	994235830,90
HL2000_4	1313472916,39	1773920874,42	1333532360,89	1212644493,95
HL2000_6	1449311132,19	1808197355,54	1468011830,95	1397981055,76
HL2000_8	1565862566,28	1823804529,44	1543558585,99	1539410016,51
HL3000_2	3894519518,55	3894519518,55	2517872968,14	2297986063,26
HL3000_4	2996287378,19	4020591440,33	2855659077,21	2733728200,60
HL3000_6	3251840168,18	4096406415,50	3160118929,20	3107582192,38
HL3000_8	3503512659,19	4134720415,71	3439158487,58	3392125649,65

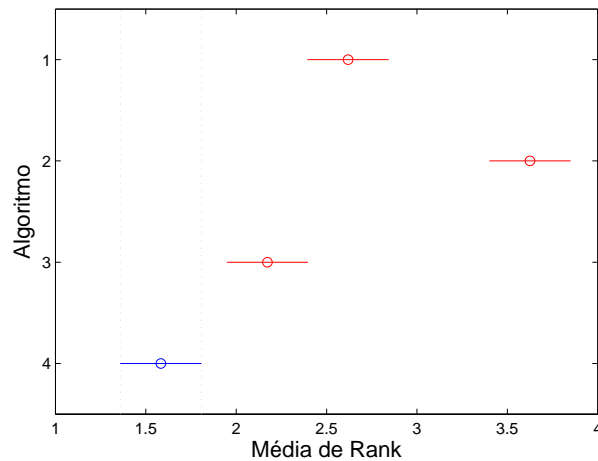


Figura 38 – Gráfico contendo informação de média e intervalo de confiança em relação ao Rank, obtido a partir do teste de Friedman e do procedimento de comparações múltiplas.

possível verificar que para as instâncias HL300\_4, HL400\_6, HL1000\_2 e HL2000\_4 o método ILS-RVND1 se destaca em comparação aos demais. O tempo de convergência é significativamente menor, e as probabilidades para alcançar a solução alvo mais rápido é de 100%, ou seja,  $P(\text{ILS-RVND1} \leq (\text{AG}, M)) = 100$ . Isso acontece porque o método de construção já atinge o valor alvo para estas duas instâncias.

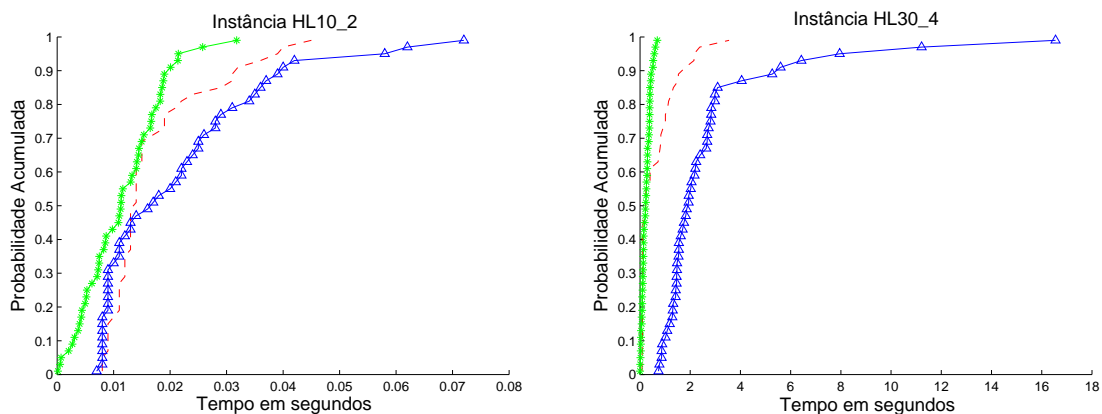


Figura 39 – Gráficos TTT.

## 5.7 Conclusão

O projeto de rede com do tipo E-R não capacitado com alocação múltipla foi abordado neste capítulo. Para resolução deste problema novamente foram utilizadas as duas heurísticas paralelas propostas neste trabalho, e o desempenho da melhor versão comparado com outros algoritmos da literatura.

Inicialmente foi verificado o desempenho de adaptações das heurísticas construtivas aplicada ao problema USAHLP para o problema UMAHLP. O desempenho do método  $M1$  em relação a  $M2$  foi ligeiramente superior, então foi optado em usar este método para gerar soluções. Com o objetivo de acelerar este procedimento, foi proposto uma variante deste método que define somente parte dos nós como candidato a concentradores. As duas variantes foram comparadas, e escolhida a que apresentou melhores resultados em relação a qualidade de solução  $\times$  tempo computacional. Neste caso foi optado em utilizar a construção com restrição de candidatos a concentradores.

Para refinar as soluções foram utilizadas as ideias aplicadas ao USAHLP, com as devidas alterações para tratar as particularidades do problema tratado neste capítulo. A vizinhança troca de alocação não foi utilizada pois no UMAHLP é considerado, a priori, que todos os nós não concentradores estão alocados a todos os concentradores. Os métodos ILSRVND1 e ILSRVND2 foram comparados, e nesta variante os resultados dos indicadores mostram que ILS-RVND1 obteve desempenho um pouco melhor que ILS-RVND2.

Neste caso houve uma inversão em relação ao algoritmo escolhido para tratar a variante USAHLP. Isso pode ser explicado devido ao tempo de processamento disponibilizado ser diferente. O algoritmo ILS-RVND2, de forma geral, converge mais rápido que ILS-RVND1. Por outro lado, este algoritmo fica preso em ótimos locais. Ao verificar os recursos que permitiram este comportamento para o ILS-RVND1, foi entendido que a definição de mais *threads* de busca permite uma melhor exploração do espaço, isso devido ao fato da segmentação das soluções para aplicar o método de perturbação, e de um maior número de soluções presentes na memória compartilhada para aplicar o método PR. Por outro lado, para intervalos de tempo menores que os utilizados neste capítulo, o ILS-RVND2 é uma boa opção.

Por fim, a comparação com os métodos da literatura mostraram que a versão ILS-RVND1 teve desempenho superior dentro do tempo disponibilizado para os testes. Os indicadores mostraram a superioridade, confirmada a partir do teste estatístico empregado. Novamente foi possível verificar que a paralelização trouxe benefício, bem como a eficiência das heurísticas propostas que não precisaram de nenhuma calibração para este problema específico, uma vez que foi utilizado os parâmetros definido no problema tratado anteriormente.

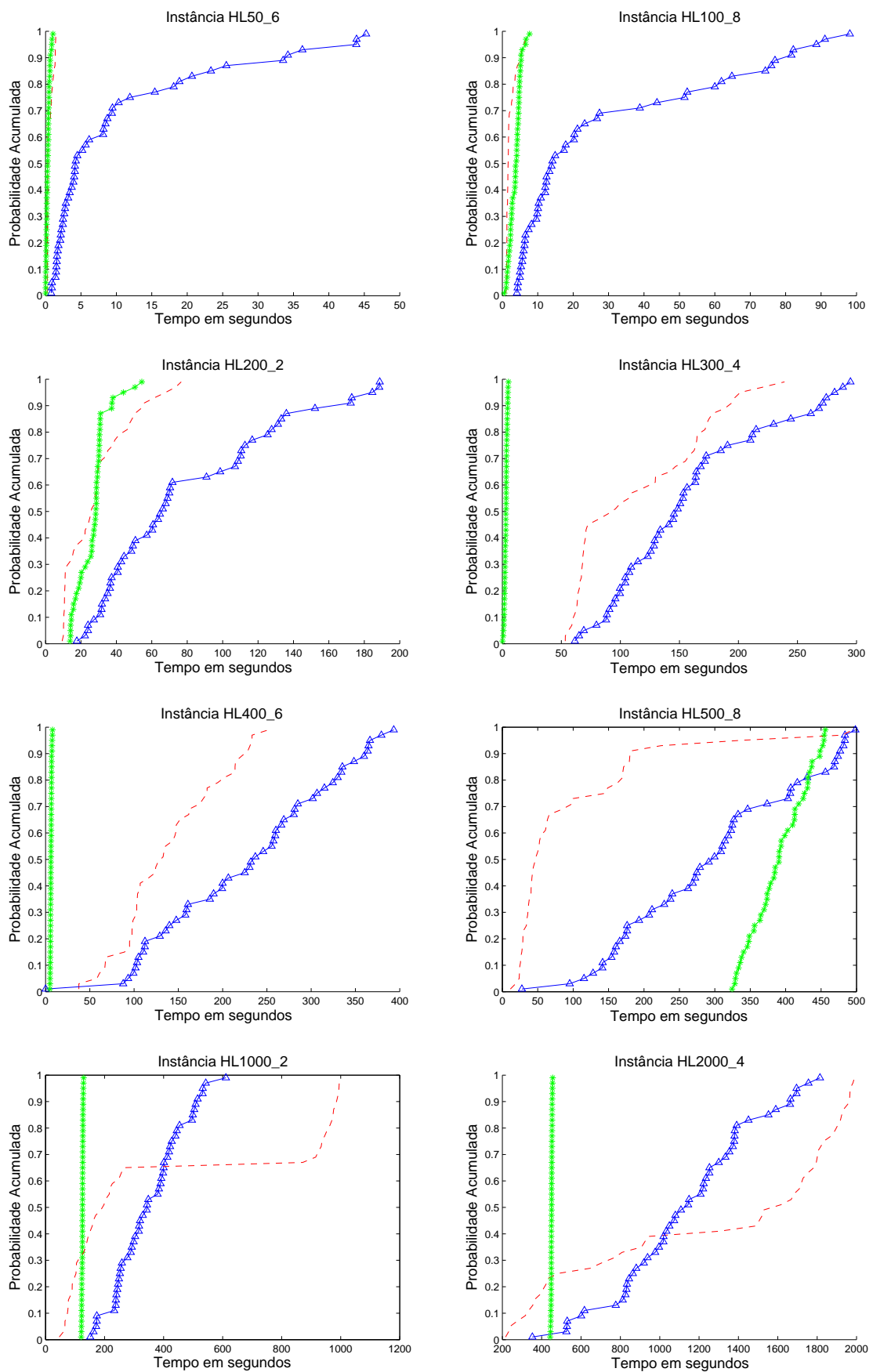


Figura 40 – Gráficos TTT.



## 6 Redes do tipo Eixo-Raio em Anel com Alocação Simples

O problema de rede do tipo E-R em anel com alocação simples pode ter diferentes definições, a que será abordada neste trabalho é baseada nas seguintes: Seja um conjunto  $N$  de nós de demanda e um conjunto  $K$  de nós candidatos a concentradores, onde  $n = |N|$ . Normalmente  $K \subseteq N$ , considerando que todos os nós de demanda são candidatos a concentradores, portanto  $K \equiv N$ . Para todo par de nós  $i$  e  $j$  ( $i, j \in N : i \neq j$ ),  $w_{ij}$  representa o fluxo de demanda entre o nó  $i$  (origem) e o nó  $j$  (destino) que é roteado por um ou mais concentradores. Normalmente  $w_{ij} \neq w_{ji}$ . Sendo  $c_{ij}$  o custo de transporte por unidade de fluxo vindo do nó  $i$  ao nó  $j$  que pode ser decomposto em três parcelas:  $c_{ij} = c_{ik} + \alpha c_{path} + c_{mj}$ , onde  $c_{ik}$  e  $c_{mj}$  é o custo de transporte do nó  $i$  até o concentrador  $k$  e do concentrador  $m$  até o nó  $j$  respectivamente. O custo de transporte  $\alpha c_{path}$  entre os concentradores, é um caminho ligando os concentradores  $k$  e  $m$ . Neste problema não existe uma aresta entre todos os concentradores, eles estão ligados em forma de uma rede em anel. Todo o custo de transporte entre concentradores é ponderado por um fator  $\alpha$  que visa representar o fator de desconto, ou economia de escala, entre os nós concentradores. Se apenas um concentrador é utilizado, isso significa que  $k = m$ , nenhum fator de desconto é utilizado. Nesse problema o número de concentradores é conhecido a priori.

Em [Contreras, Tanash e Vidyarthi \(2016\)](#) esse problema é formulado como:

$$\min \sum_i \sum_k (c_{ik} O_i + c_{ki} D_i) z_{ik} + \sum_i \sum_k \sum_{m \neq k} \alpha c_{km} x_{ikm} \quad (6.1)$$

$$\sum_k z_{ik} = 1 \quad \forall i \in N \quad (6.2)$$

$$\sum_k z_{kk} = p \quad (6.3)$$

$$\sum_k \sum_m y_{km} = p \quad (6.4)$$

$$\sum_{k < m} y_{km} + \sum_{k > m} y_{mk} = 2z_{kk} \quad \forall k \in N \quad (6.5)$$

$$O_i z_{ik} + \sum_{m \neq k} x_{imk} = \sum_{m \neq k} x_{ikm} + \sum_m w_{im} z_{mk} \quad \forall i, k \in N; k \neq i \quad (6.6)$$

$$z_{km} + y_{km} \leq z_{mm} \quad k, m \in N; m > k \quad (6.7)$$

$$z_{mk} + y_{km} \leq z_{kk} \quad k, m \in N; m > k \quad (6.8)$$

$$x_{ikm} + x_{imk} \leq O_i y_{km} \quad i, k, m \in N; m > k \quad (6.9)$$

$$x_{ikm} \geq 0 \quad i, k, m \in N \quad (6.10)$$

$$z_{km}, y_{km} \in 0, 1 \quad k, m \in N \quad (6.11)$$

em que  $O_i$  e  $D_i$  representam o fluxo originado e destinado ao nó  $i$ , respectivamente.  $y_{km}$  é uma variável que assume valor 1 quando existe uma conexão entre os concentradores  $k$  e  $m$ , e 0 caso contrário. A função objetivo (6.1) é composta pelo custo de fluxo entre os nós não concentradores e seus concentradores (primeiro termo) mais o custo de fluxo entre concentradores (segundo termo). As restrições (6.2) asseguram que cada nó será atribuído a apenas um concentrador. A restrição (6.3) indica o número de concentradores que devem ser instalados. A restrição (6.4) garante que o número de arcos deve ser igual ao número de concentradores  $p$ . As restrições (6.5) asseguram que cada concentrador deve estar conectado a exatamente dois concentradores. As restrições 6.6 garantem a conservação do fluxo. As restrições (6.7) e (6.8) garantem que os arcos definidos entre concentradores só seja ativado caso os nós das extremidades sejam concentradores. As restrições (6.9) assegura que o fluxo entre os concentradores deve ser feito através do ciclo configurado. Por fim, as restrições (6.10) e (6.11) referenciam o domínio das variáveis do problema.

Um exemplo de uma rede do tipo Eixo-Raio é mostrado na Figura 41, onde os concentradores estão representados como triângulos, os pontos de demanda como círculos e as alocações são representadas pelas linhas, bem como as conexões entre os concentradores. Observa-se que as alocações entre os nós de demanda (não concentradores) e concentradores são simples. Além disso, os concentradores estão ligados em forma de anel.

Uma potencial aplicação onde a localização de concentradores em ciclo é necessário surgiu na concepção de redes de telecomunicações. Neste caso, as instalações de concentradores correspondem a equipamentos eletrônicos, como *hubs*, multiplexadores e comutadores, e os fluxos de demanda são pacotes de dados encaminhados por uma variedade de meios físicos, tais como diferentes tipos de cabos ou através do ar, utilizando canais por satélite. Também podem ser aplicadas ao transporte público, onde as vias rápidas poderiam ser representadas pela organização dos concentradores em ciclo, enquanto as conexões aos



Figura 41 – Exemplo de uma rede do tipo E-R com alocação simples e concentradores ligados em anel

concentradores as vias alimentadoras para os bairros. Tal configuração pode ser encontrada em cidades como Beijing (China), Glasgow (Escócia) e Paris (França).

Esse problema foi proposto por [Contreras e Fernández \(2012\)](#) e ainda não é muito explorado na literatura, embora existam vários problemas similares onde uma rede de concentradores em formato de anel é solicitada. Por exemplo, [Labbé e Yaman \(2004\)](#) introduziram o problema denominado anel-estrela, do inglês *Ring Star Problem* (RSP), onde o objetivo é localizar os concentradores, de forma que a conexão entre eles esteja em formato de anel, minimizando o custo de instalação da rede e das atribuições dos nós não concentradores aos concentradores. Neste trabalho é proposto um algoritmo *Branch-and-Cut* junto a uma formulação eficiente para o problema. Tal metodologia permitiu que fossem resolvidas instâncias de até 300 nós. [Grendeau, Laporte e Semet \(1997\)](#) estudaram uma variação deste problema no qual todos os nós devem estar dentro de uma distância máxima do ciclo.

[Current e Schilling \(1994\)](#) propuseram uma heurística multiobjetivo para uma variação do problema de ciclo mediano. Neste trabalho o objetivo foi localizar uma rede em ciclo com  $p$  nós, minimizando o custo de instalação do ciclo e o custo de atribuição dos outros nós a este ciclo. Foi utilizada uma instância com 681 nós, onde 70 nós foram identificados como candidatos a concentradores. Um algoritmo aproximativo é proposto por [Kuroki e Matsui \(2017\)](#), neste trabalho apenas é apresentada a metodologia empregada, nenhum resultado é apresentado.

[Labbe et al. \(2005\)](#) estudaram outro problema similar, denominado problema do ciclo mediano, do inglês *Median Cycle Problem*. Este problema coloca-se na concepção de infra-estruturas em forma de anel e consiste em encontrar um ciclo simples que minimiza os custos de instalação do ciclo, de modo a que o custo de transmissão total dos nós não pertencentes a esse ciclo não excedam uma determinada restrição orçamental. Nesse trabalho também é proposto um algoritmo *Branch-and-Cut*, os resultados para um conjunto de instância da literatura mostrou a eficiência da metodologia.

Segundo [Contreras, Tanash e Vidyarthi \(2016\)](#), até o momento é conhecido apenas o trabalho proposto por eles para resolução do problema de alocação de concentradores em ciclo, de acordo com a formulação apresentada. Nesse trabalho foram propostos um método exato e uma heurística para resolver o problema. A heurística é composta por duas fases, construção e busca local. Para a fase de construção é utilizado uma fórmula para definir potenciais concentradores, que serão escolhidos de uma forma semi-gulosa de acordo com a sua aptidão. Após a escolha dos concentradores uma rede em anel é criada utilizando o critério do vizinho mais próximo. Para a fase de busca local é utilizado um algoritmo baseado no método VND. Quatro diferentes tipos de vizinhança são utilizados: *swap*, que visa trocar a atribuição de dois nós não concentradores; *shift*, consiste em trocar um nó não concentrador de concentrador; troca de função, é feita uma troca de funções entre nós concentradores e não concentradores, posteriormente os nós não concentradores são atribuídos ao concentrador mais próximo; 2-opt, modifica as ligações da rede em anel trocando duas arestas da rede. Para o método exato, foi proposto um algoritmo *Branch-and-Cut* que usa uma formulação baseada em fluxo e um procedimento para obtenção de limites inferiores para a árvore de enumeração (*Branch-and-Bound*). Ambos os métodos foram testados utilizando instâncias de até 100 nós, e obtiveram bom desempenho.

Outros trabalhos correlacionados podem ser visto em [Contreras, Tanash e Vidyarthi \(2016\)](#), [Lee, Chiu e Sanchez \(1998\)](#), [Baldacci, Amico e Gonzalez \(2007\)](#).

## 6.1 Metodologia

Nesta seção, será detalhada a heurística proposta para resolver o problema abordado neste capítulo. Novamente serão utilizadas as variantes do método ILS-RVND-PR.

## 6.2 Representação Computacional

Para representar uma solução, são utilizadas as ideias de representações dos dois problemas tratados anteriormente. O vetor  $S^A$  contém a informação da alocação dos nós, ou seja, a qual nó concentrador o  $i$ -ésimo nó, representado pelas posições do vetor, está alocado.  $O_s$  um vetor contendo quais nós são concentradores. *path* contém a informação de como está configurada a conexão entre os concentradores e serve como indicador para a matriz  $M$ . O indicador mapeia cada linha da matriz  $M$  sendo um concentrador instalado. Além disso, como está sendo tratado o problema de alocação de concentradores em anel, o primeiro e último nó deste vetor estão conectados. A matriz  $M$  auxilia a identificação de um caminho entre cada par de concentradores. O conteúdo da posição  $i, j$  da matriz  $M$  identifica para qual nó um fluxo que está localizado no nó  $i$  deve seguir para alcançar o nó  $j$ , utilizando o menor caminho.

Assim sendo, considere a disposição dos nós conforme a Figura 42. Para alcançar o nó 6 partindo do nó 3 é necessário encontrar primeiro em quais concentrados estes nós estão alocados. Esta informação é obtida no vetor  $S^A$ , neste caso temos que os nós 3 e 6 estão alocados, respectivamente, nos nós concentradores 5 e 2. Estes concentradores, estão, respectivamente, nas posições 3 e 1 do vetor  $path$ . Assim, para obter um caminho entre estes concentradores é necessário verificar a posição  $m_{31} \in M$ . O valor encontrado nesta posição indica para qual concentrador o fluxo deve se deslocar para alcançar seu destino. Neste caso, o valor de  $m_{31} = 4$ . Novamente é feito o mesmo procedimento, porém dessa vez utilizando o concentrador 4 como origem. De forma genérica, o procedimento é repetido até que  $m_{ij} = \text{destino}$ .

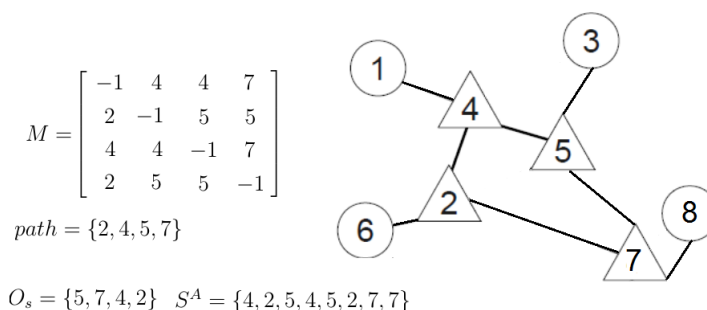


Figura 42 – Representação computacional para o problema de localização de concentradores em anel com alocação simples.

### 6.2.1 Avaliação da Solução

A avaliação de uma solução pode ser feita de duas formas, parcial ou total. Quando é feita apenas uma mudança de alocação, ou seja, um nó  $i \in C_s$  é alocado a outro concentrador, é calculado apenas o impacto nos custos de transporte para aquele nó, assim como foi feito no USAHLP, equações 14 e 15. A diferença é que neste problema o número de concentradores entre os nós de origem e destino pode ser maior que 2.

Quando é feita a troca de concentradores, ou seja, a abertura de um novo concentrador e o desligamento de um antigo, a função objetivo e  $M$  são totalmente recalculadas, utilizando o algoritmo de Dijkstra para atualizar  $M$ . Além disso, para obter a melhor configuração, em anel, entre os concentradores, foi utilizado o algoritmo Linkern, uma versão do algoritmo *Lin-Kernighan* (LIN; KERNIGHAN, 1973), disponível no pacote Concorde (APPLEGATE et al., 1996). Este consiste em um *solver* específico para problemas envolvendo o Problema do Caixeiro Viajante simétrico, escrito em C e disponível gratuitamente para uso acadêmico.

## 6.2.2 Método de Construção

O método consiste em definir 30% dos nós de uma determinada instância como concentradores, e posteriormente remove-los até que o número de concentradores seja igual a  $p$ . Assim sendo, inicialmente  $O_s = V$  e  $C_s = \emptyset$ , onde  $N \subset V$ . A cada iteração, para cada  $j \in O_s$  é mudado o papel de concentrador para não concentrador, e recalculada a função objetivo. Neste processo os nós alocados ao concentrador  $j$  são realocados para um novo concentrador  $i \in O_s$ . A definição de  $V$  é feita assim como no capítulo 6, baseado no fluxo de origem e destino dos nós. Caso  $p > |V|$ , a quantidade de elementos em  $V$  passa a ser igual  $p$ . Este procedimento (ver algoritmo 3) é baseado no método proposto para o problema USAHLP. Inicialmente são definidos os conjunto dos nós concentradores e não concentradores (linhas 2 e 3). Posteriormente, é feita a busca do concentrador que se retirado tem um menor impacto na função objetivo (linha 5). As linhas 6 e 7 mostra a atualização da solução corrente. O método termina quando o número de concentradores é igual ao número de concentradores que precisam ser localizados no problema (linha 8).

---

**Algoritmo 6:** MÉTODO DE CONSTRUÇÃO PARA O PROBLEMA DE PROJETO DE REDES DO TIPO EIXO-RAIO EM ANEL COM ALOCAÇÃO SIMPLES.

---

```

1 início
2    $O_s \leftarrow \{V\}$ 
3    $C_s \leftarrow N \setminus V$ 
4   repita
5      $j \leftarrow \arg \min_{j \in C_s} \{\delta'_j\}$ 
6      $C_s \leftarrow C_s \cup \{j\}$ 
7      $O_s \leftarrow O_s \setminus \{j\}$ 
8   até  $|V| = p$ ;
9 fim
```

---

## 6.2.3 Estruturas de Vizinhança

Duas estruturas de vizinhança foram utilizadas para o problema abordado neste capítulo, denominadas "Troca de Função" e "Realocação". Estas estruturas são descritas a seguir:

- Troca de Função: cada concentrador alocado a  $k$  é escolhido para se tornar um concentrador. Em seguida, os nós não concentradores que estavam alocados a  $k$  são relocados para o concentrador mais próximo. O número de vizinhos gerados a partir dessa vizinhança é  $|C_s|$ , e a quantidade de operações (no pior caso) para avaliar todas as soluções é aproximadamente  $|C_s|(|O_s|^2 + n^2)$ .
- Realocação: para cada nó  $j \in C_s$  é feita a tentativa de realoca-lo a um concentrador  $k \in O_s$ . O número de vizinhos gerados a partir dessa vizinhança é  $|C_s|$ , e a quantidade

de operações (no pior caso) para avaliar todas as soluções é aproximadamente  $2n|C_s|(|O_s| - 1)$ .

Além disso, foi usado o *solver* Concorde, mais especificamente o algoritmo Linkern, para definir o ciclo entre os concentradores. O algoritmo ILS-RVND utiliza as duas estruturas propostas mais o Concorde na fase de intensificação, em outras palavras, como recursos para a busca local.

#### 6.2.4 Perturbação

Conforme explicado no Capítulo 2, o procedimento de perturbação utilizado neste trabalho restringe a cada *thread* de busca algumas posições do vetor solução para aplicar a perturbação. Os movimentos utilizados para perturbar uma solução são detalhados a seguir:

- Troça de Função: Este movimento pode alterar a estrutura de uma solução mudando a função de dois nós. Assim, um nó não concentrador passa a ser concentrador, enquanto um nó concentrador passa a ser um nó não concentrador. Após a troca de função, todos os nós não concentradores são alocados aos concentradores mais próximos.
- Realocação: Este movimento altera a estrutura de uma solução mudando a alocação dos nós não concentradores aleatoriamente.

Além disso, a variável  $\pi$  define a probabilidade de cada posição de uma solução sofrer alteração. Quanto maior este valor, maior a probabilidade de um dos movimentos ser executado. Os movimentos são escolhidos aleatoriamente.

#### 6.2.5 Path-Relinking

O método PR foi adaptado para que a cada iteração fossem geradas soluções viáveis para o problema, ou seja, onde o número de concentradores fosse igual a  $p$ . Para isso, considere as soluções guiada  $s'$  e guia  $s''$ . O primeiro passo consiste em obter quais concentradores estão instalados em  $s''$  e que não estão instalados em  $s'$ , ou seja,  $D = O_{s''} \setminus O_{s'}$ . Considerando  $O_{s'} = \{2, 4, 5\}$ ,  $O_{s''} = \{1, 2, 4\}$ , conforme apresentado nas colunas (a) e (c), respectivamente, na Figura 43, temos  $D = \{1\}$ . Cada concentrador da solução guiada é trocado por um concentrador do conjunto  $D$ , e a melhor solução entre as geradas torna-se a solução corrente para nova verificação. O processo se repetirá até que a solução corrente e a solução guia sejam iguais em relação aos concentradores. A coluna (b) da Figura 43 ilustra as soluções intermediárias geradas pelo método PR. Caso a primeira ou segunda linha da coluna (b) fosse escolhida como solução corrente,  $D$  seria recalculado,

novas soluções seriam geradas e uma nova solução corrente seria escolhida. No entanto, se a terceira linha fosse escolhida, o método seria encerrado, já que os concentradores são iguais aos da solução guia.

Para evitar ciclagem, ou seja, que soluções anteriormente definidas como solução corrente no PR sejam exploradas mais uma vez, é utilizada uma lista  $L$  que armazena a configuração destas soluções. Assim, todas as soluções contidas em  $L$  são proibidas de serem escolhidas em uma mesma chamada do método. Por exemplo, baseado na Figura 43, considere que a partir da solução corrente, ilustrada na coluna (a), foram geradas as soluções da coluna (b). Escolhendo a segunda solução desta coluna como a melhor gerada a partir da solução corrente, teríamos  $D = \{4\}$  e  $L = \{(2, 4, 5), (1, 2, 5)\}$ . Logo, as seguintes soluções poderiam ser geradas a partir da nova solução corrente, com o objetivo de chegar a solução guia,  $\{(2, 4, 5), (1, 4, 5), (1, 2, 4)\}$ . Deste conjunto, a solução composta pelos concentradores 2,4 e 5 não é permitida, já que está em  $L$ .

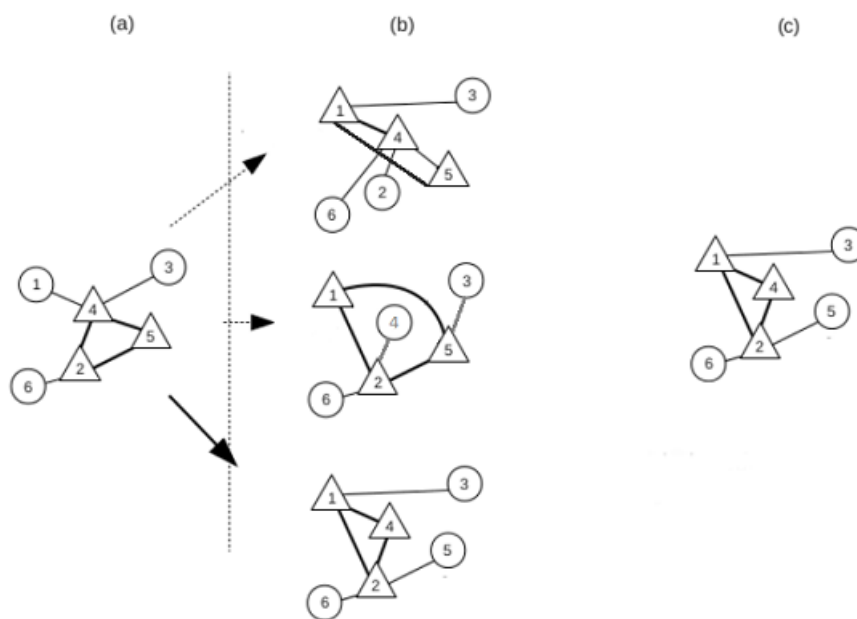


Figura 43 – Ilustração do método *Path-Relinking* aplicado ao problema de localização de concentradores em anel com alocação simples.

### 6.3 Experimentos Computacionais

Esta seção apresenta os resultados obtidos pelos métodos propostos neste trabalho e um algoritmo da literatura. Novamente o computador *PC3* foi utilizado, o qual contém dois processadores Intel Xeon E5-2620 de 2.1 Ghz, cada um contendo 6 núcleos totalizando 24 *threads*. Além disso, a máquina possui 64 Gb RAM e executa um sistema operacional baseado em Unix (Linux).

Inicialmente é feita uma breve análise em relação ao desempenho dos diferentes métodos de construção, proposto neste trabalho e o utilizado por [Contreras, Tanash e Vidyarthi \(2016\)](#), atual estado da arte. Posteriormente são utilizados alguns indicadores e uma análise estatística para comparar o desempenho dos métodos ILS-RVND1, ILS-RVND2, bem como a heurística proposta por [Contreras, Tanash e Vidyarthi \(2016\)](#).

O experimento foi configurado da seguinte maneira: foram utilizadas 56 instâncias (AP) e seis diferentes valores para  $p = \{3, 4, 5, 6, 7, 15\}$  em 10 execuções distintas. Para  $p = 15$  somente instâncias onde  $n \geq 50$  foram consideradas. Como critério de parada foi definido o tempo máximo de processamento ( $n$  segundos).

A Figura 44 mostra os valores de função objetivo e o tempo demandado por cada método de construção, para diferentes valores de  $p$ .

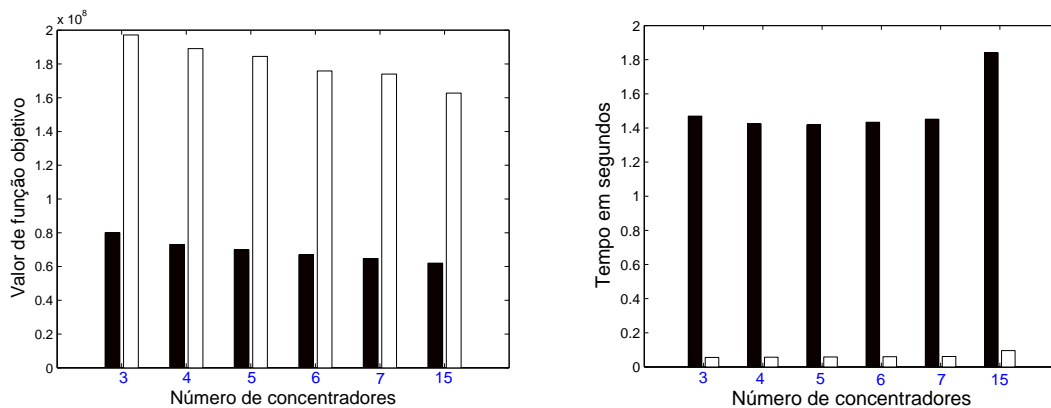


Figura 44 – Função objetivo obtida e tempo demandado para construir uma solução utilizando diferentes métodos. Construção proposta (Preto) × Construção da Literatura (Branco).

É possível perceber que a qualidade da solução obtida com o método proposto neste trabalho é superior, sendo, em média, 60% menor a função objetivo. Embora o tempo demandado pelo método de construção proposto seja maior, o tempo demandado para instância de 200 nós não ultrapassa os dois segundos, o que viabiliza o uso desta abordagem. A diferença de qualidade entre as soluções pode ser explicada devido o método da literatura considerar apenas o fluxo para definir os concentradores iniciais. Neste trabalho, o fluxo apenas serve para definir concentradores em potencial, mas é o custo associado a cada um deles, em relação a distribuição do fluxo, que o transforma, ou não, em concentrador.

Os valores médios de função objetivo obtidos pelos métodos ILS-RVND1, ILS-RVND2 e GRASP ([CONTRERAS; TANASH; VIDYARTHI, 2016](#)), agrupados pelos valores de  $p$ , são apresentados na Figura 45. É possível observar que a proposta desse trabalho obtém melhores valores de função objetivo em todos os casos.

O comportamento apresentado pelo método GRASP foi o esperado, isso porque os tempos de convergência reportados por [Contreras, Tanash e Vidyarthi \(2016\)](#) são

significativamente mais altos que  $n$ , tempo disponibilizado para processamento neste trabalho. Além disso, os resultados obtidos por esses autores mostram que o número de concentradores impacta no tempo de convergência do método. Como pode ser observado na Figura 45, o pior desempenho foi para instância onde  $p = 15$ .

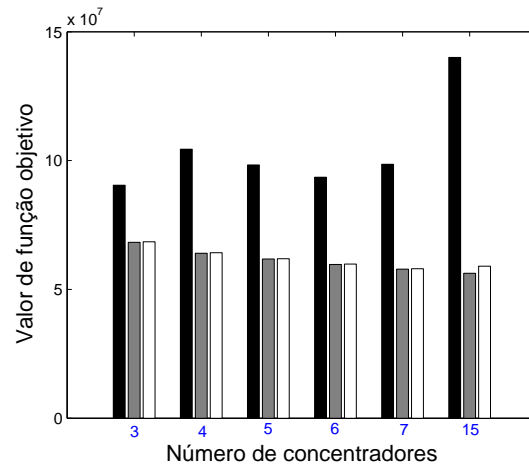


Figura 45 – Função objetivo obtida pelos diferentes métodos utilizados neste trabalho. Métodos: GRASP (Preto), ILS-RVND1 (Cinza) e ILS-RVND2 (Branco).

Avaliando somente instâncias onde  $n \leq 100$ , para o qual o GRASP foi concebido, o comportamento deste método (GRASP) é melhor. Isto pode ser observado na Figura 46, onde a diferença entre os métodos é significativamente menor, considerando apenas as instâncias onde  $p \leq 7$ . Mesmo para instâncias onde  $n \leq 100$ , quando  $p = 15$ , o GRASP não mantém o mesmo desempenho.

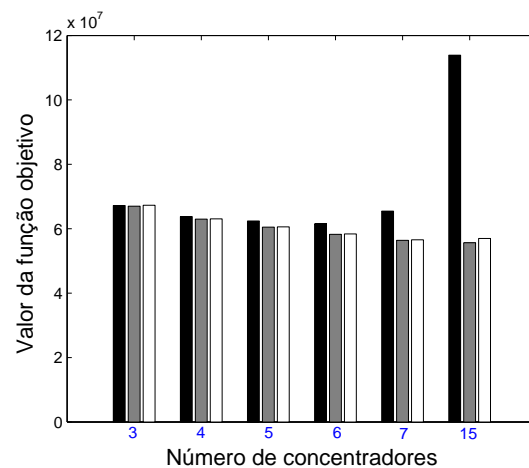


Figura 46 – Função objetivo obtida pelos diferentes métodos utilizados neste trabalho para  $n \leq 100$ . Métodos: GRASP (Preto), ILS-RVND1 (Cinza) e ILS-RVND2 (Branco).

Os indicadores DevMin, DevMed e KBS foram computados para avaliar o desempenho dos três métodos. A tabela 17 mostram os resultados obtidos, ao todo foram considerados 3200 execuções de cada método. Em 90% dos casos o método ILS-RVND1

obteve a melhor solução entre os algoritmos, enquanto o método GRASP alcançou apenas em 24%. Embora o ILS-RVND2 tenha obtido menos soluções "KBS", a partir dos indicadores DevMed e DevMin é possível verificar que as soluções encontradas por este método estão próximas das melhores soluções encontradas. O algoritmo GRASP apresentou comportamento ruim para todos os indicadores.

Tabela 17 – Valores de DevMin, DevMed e KBS obtidos para os métodos GRASP, ILS-RVND1 e ILS-RVND2

Indicadores	ILS-RVND1	ILS-RVND2	GRASP
KBS	2892	1123	795
DevMin	0	0,0006	0,632
DevMed	0	0,0008	0,650

Ao aplicar o teste estatístico de Friedman e o procedimento de múltiplas comparações, o resultado apontou, com o nível de significância igual a 0,05, que os algoritmos apresentam diferença significativa. Além disso, o procedimento de múltiplas comparações mostra que o método ILS-RVND1 é o melhor, como pode ser observado na Figura 47.

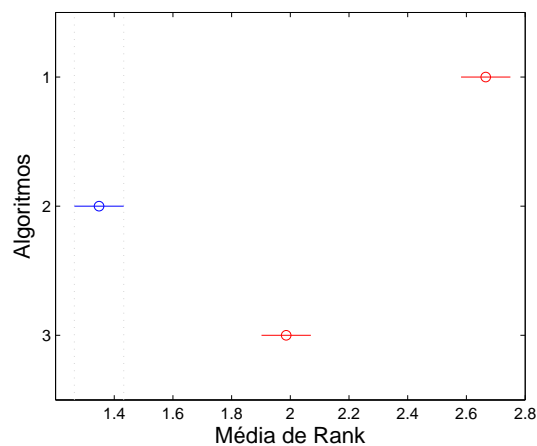


Figura 47 – Gráfico contendo informações de média e intervalo de confiança em relação ao Rank, obtido a partir do teste de Friedman e do procedimento de comparações múltiplas. Algoritmo: 1 - GRASP; 2 - ILS-RVND1; 3 - ILS-RVND2

Alguns recursos utilizados pelas heurísticas propostas podem explicar a diferença de desempenho. Ao computar o tempo gasto pelo movimento 2-opt, e as soluções obtidas a partir desse movimento, verificamos que a utilização do solver Concorde foi mais eficiente. Além disso, a partir das soluções obtidas pelo método de construção proposto, é possível alcançar ótimos locais de qualidade rapidamente, o que não acontece com o GRASP. De acordo com a literatura, para que o algoritmo GRASP possa ter um bom desempenho, três características devem ser observadas: tempo demandado para gerar uma solução,

diversidade e qualidade. Neste caso, conforme mostrado anteriormente, as soluções obtidas pelo método de construção GRASP apresentam baixa qualidade.

Além disso, como esperado, o paralelismo mais uma vez favoreceu a convergência dos métodos, permitindo assim uma exploração mais ampla no espaço de busca. Não menos importante, a tática de diversificação e o método PR contribuirão efetivamente na obtenção de boas soluções. As Figuras 48 e 49 apresentam gráficos TTT para algumas instâncias, sendo os métodos ILS-RVND1, ILS-RVND2 e GRASP representados, respectivamente, pelos marcadores triângulo, asterisco e círculo. Como solução alvo foi definida a pior solução encontrada entre os três métodos.

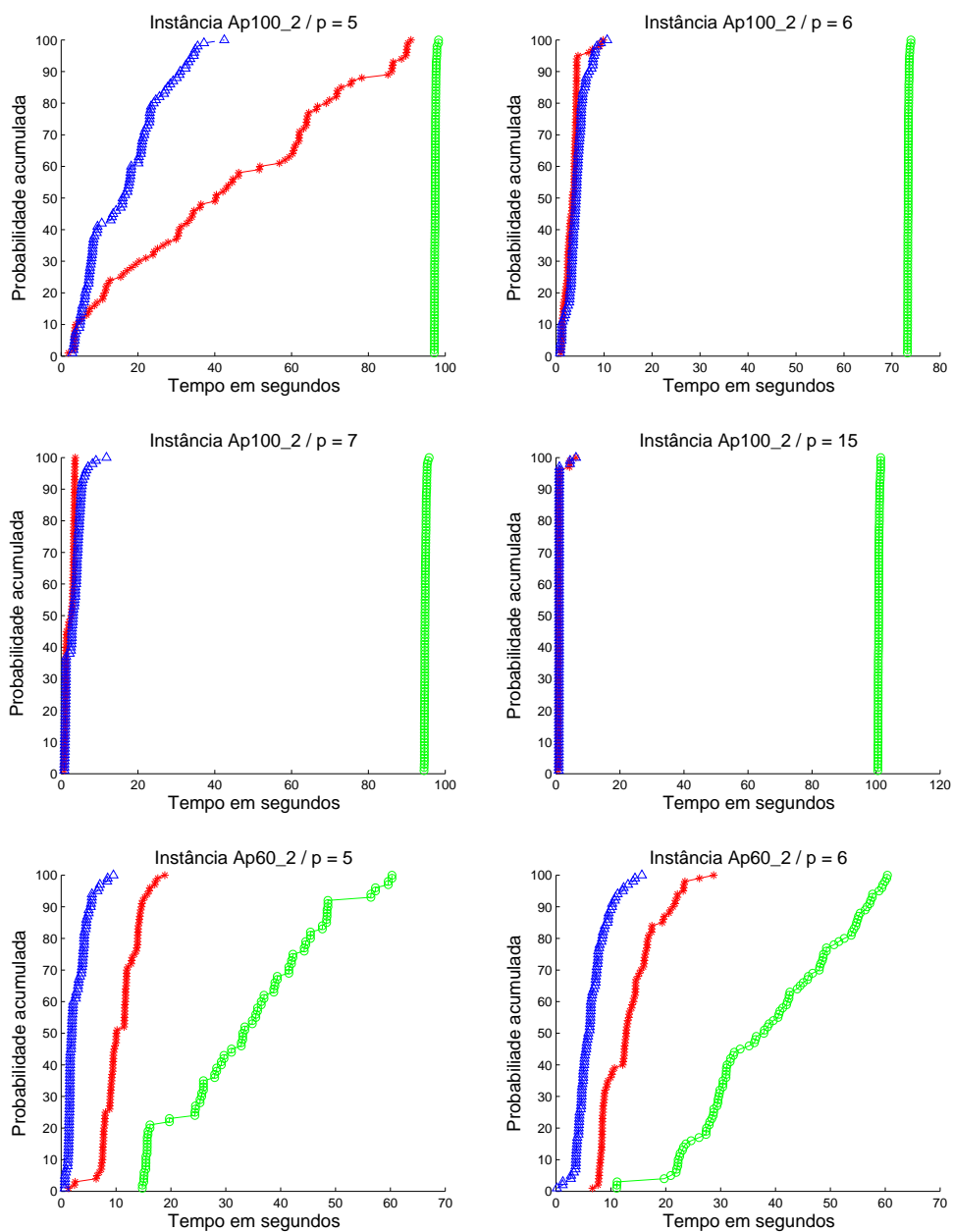


Figura 48 – Gráficos TTT.

A probabilidade de encontrar estas soluções, em menos tempo, é significamente

maior para os métodos propostos neste trabalho. Além disso, no geral, o comportamento do método ILS-RVND2 se mantém o mesmo, quando comparado sua aplicação a outras variantes do problema de localização de concentradores. Embora sua convergência seja mais rápida, o algoritmo fica preso em ótimos locais. Este comportamento faz com que este algoritmo seja eficiente para ser usado na obtenção de soluções razoáveis quando é disponibilizado pouco tempo de processamento.

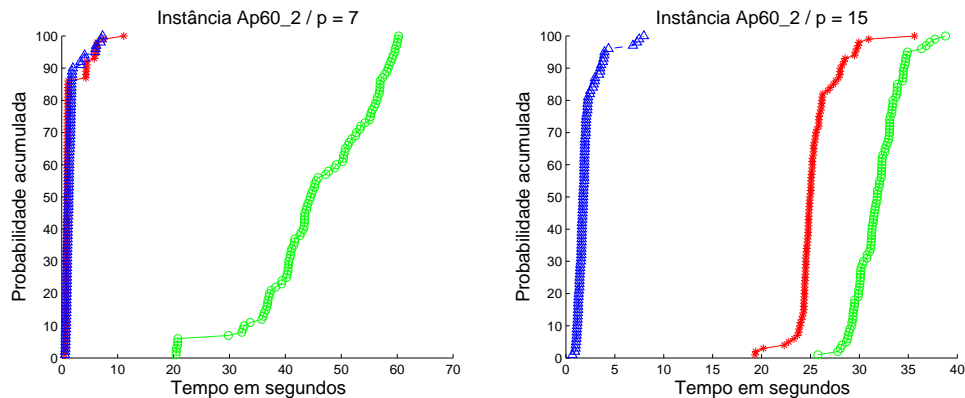


Figura 49 – Gráficos TTT.

## 6.4 Conclusão

Neste capítulo, o projeto de redes do tipo E-R em anel com alocação simples foi tratado. Para resolução dos problemas novamente foram utilizadas duas heurísticas paralelas propostas neste trabalho. O desempenho destes algoritmos foram comparados com um algoritmo heurístico referência da literatura.

Inicialmente, foi comparado o desempenho das diferentes heurísticas construtivas, uma proposta neste trabalho e outra baseada no algoritmo referência. Os resultados mostraram que a qualidade das soluções obtidas pelo método aqui proposto é significativamente mais eficiente, gerando soluções de qualidade com um pequeno esforço computacional. Não menos importante, vale ressaltar que a ideia deste método é a mesma utilizada para as outras variantes, sendo apenas adaptado o critério de parada.

Para refinar as soluções, foram utilizadas algumas vizinhanças para resolver o USAHLP e o solver concorde, responsável em gerar rotas entre os concentradores de forma eficiente. Além disso, foi proposto um método PR que a cada iteração é capaz de gerar soluções válidas para o problema.

Três métodos foram comparados, sendo dois propostos neste trabalho e o método GRASP encontrado na literatura. Entre os algoritmos, aquele que apresentou melhor desempenho foi o ILS-RVND1. O comportamento dos algoritmos propostos neste capítulo, em relação ao problema UMAHLP, foi parecido. O método ILS-RVND2 é mais rápido para

alcançar determinadas soluções, porém os resultados mostraram que o resultados finais alcançados pelo ILS-RVND1 são melhores. Isso evidencia uma deficiência deste método, que embora alcance soluções de qualidade em um curto espaço de tempo, fica estagnado em ótimos locais de pior qualidade.

Foi verificado mais uma vez a eficiência dos métodos propostos em relação a principal proposta heurística da literatura. De forma geral, é possível afirmar que a heurística utilizada é robusta, uma vez que foram necessárias apenas algumas adaptações para obter resultados de qualidade. Tais adaptações foram necessárias devido o problema abordado neste capítulo definir, a priori, o número de concentradores.

## 7 Conclusões Finais e Trabalhos Futuros

É notório que nos últimos anos houve uma popularização de computadores pessoais equipados de processadores com múltiplos núcleos. Isto tornou possível a proposta de algoritmos paralelos sem a necessidade de recursos específicos, tais como CUDA (GPU), supercomputadores ou *clusters*. Devido ao fenômeno eletromigração, que inviabiliza o aumento significativo da frequência dos novos processadores e à quantidade crescente de dados a serem analisados, o que requer cada vez mais poder de processamento, a opção de fazer o uso da computação paralela para criar algoritmos de alto desempenho vem sendo cada vez mais explorada pela comunidade científica.

Por outro lado, embora exista uma tendência para utilização deste tipo de recurso, é possível verificar na literatura que heurísticas paralelas ainda tem sido pouco exploradas. Quando se trata dos problemas estudados neste trabalho é praticamente inexistente a proposta de metodologias que fazem uso deste recurso. Este fato é reforçado, ao se escolher para comparação com a metodologia proposta, os principais algoritmos da literatura, que são sequenciais. Assim, uma das contribuições desta tese é propor heurísticas que fazem o uso deste recurso, o que pode incentivar novos pesquisadores a seguirem aperfeiçoando o uso de estratégias heurísticas de forma paralela.

Não menos importante, foram tratados neste trabalho diferentes variantes de um dos problemas mais importantes na concepção de localização de facilidades, o projeto de redes E-R. Este é um problema que tem sua devida importância na área de otimização combinatória com aplicação em diferentes contextos, como sistemas logísticos de transporte, sistemas de telecomunicações, entre outros. Foram escolhidos três problemas, entre tantos, por representarem diferentes contextos de aplicações práticas.

Em relação as heurísticas propostas, todas foram baseadas na mesma meta-heurística, denominada ILS-RVND-PR, que foram divididas em duas diferentes variantes, ILS-RVND1 e ILS-RVND2. Além disso, a concepção das heurísticas para tratar os problemas UMAHLP e *Cycle Hub*, foram baseadas na proposta para resolver o USAHLP. Isto mostrou que pequenas variações no ILS-RVND-PR podem fornecer bons resultados a diferentes problemas de alocação de concentradores. Em todos os problemas os métodos propostos foram superiores as principais heurísticas encontradas na literatura. Além disso, para problemas onde foi possível obter o ótimo a partir dos métodos exatos, o algoritmo alcançou a mesma solução significativamente mais rápido.

Outro aspecto importante a ser observado é que foram utilizadas instâncias com até 3000 nós, o que até então não vinha sendo tratado na literatura, onde comumente são encontrados problemas de no máximo 400 nós. Problemas desta escala surgem em

demandas de redes de computadores, bem como em logística. Assim, como ideia de trabalho futuro, seria interessante aplicar a ideia desta heurística a problemas de localização de concentradores envolvendo mais restrições. Um exemplo seria o planejamento da malha área brasileira levando em consideração diferentes níveis de concentradores, que neste contexto seriam aeroportos em nível regional e nacional.

Tal estudo é importante porque o Brasil, que é um país com grandes dimensões territoriais, não possui uma rede de transportes que permita o pleno desenvolvimento da economia do interior e o acesso de sua população a bens e serviços dos quais os moradores da região litorânea e das grandes capitais desfrutam. Existem projetos para proporcionar o desenvolvimento nacional e democratizar o acesso as mais diversa regiões, o que pode ser feito via transporte aéreo. A ideia é dotar o Brasil de uma rede de aeroportos regionais operando a preços competitivos, de forma a garantir que uma parcela significativa da população brasileira esteja próximas a aeroporto com condições de operar voos regulares, e com custos atrativos. Segundo um estudo da ANAC, o preço das passagens regionais é 31% mais alto, em média, do que o dos voos entre capitais, o que dificulta o acesso a população a este serviço. Segundo informações coletadas do Instituto Brasileiro de Geografia e Estatística (IBGE) e na Agência Nacional de Aviação Civil (ANAC), o Brasil conta atualmente com 5478 municípios e 618 aeroportos públicos, o que torna a resolução deste problema um desafio.

Por fim, outra possível opção de pesquisa futura seria verificar o desempenho das heurísticas desenvolvidas neste trabalho utilizando placas gráficas (GPU). Isto resultaria em um trabalho com o objetivo de identificar os diferentes aspectos relacionados ao desempenho na implementação de meta-heurísticas por meio de mecanismos que utilizam CPUs e GPUs.

## Referências

- ABDINNOUR-HELM, S. A hybrid heuristic for the uncapacitated hub location problem. *European Journal of Operational Research*, v. 2–3, n. 106, p. 489–499, 1998. Citado na página 63.
- ABDINNOUR-HELM, S. A hybrid heuristic for the uncapacitated hub location problem. *European Journal of Operation Research*, v. 2, p. 489–499, 1998. Citado na página 75.
- ABDINNOUR-HELM, S.; VENKATARAMANAN, M. Solution approaches to hub location problems. *Annals of Operations Research*, v. 78, p. 31–50, 1998. Citado na página 74.
- ABYAZI-SANI, R.; GHANBARI, R. An efficient tabu search for solving the uncapacitated single allocation hub location problem. *Computers & Industrial Engineering*, v. 93, p. 99 – 109, 2016. ISSN 0360-8352. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0360835215005082>>. Citado 4 vezes nas páginas 63, 79, 84 e 97.
- AL-SHANDAWELY, M. A. *Impacts of data structures and algorithms on multi-core efficiency*. Dissertação (Mestrado) — Royal Institute of Technology, School of Computer Science and Communication, 2010. Citado na página 89.
- ALBA, E.; NEBRO, A. J. *Parallel Meta-Heuristic: A New Class of Algorithms*. [S.l.]: John Wiley and Sons, 2005. Citado na página 36.
- ALEX, R.; RESENDE, M.; RIBEIRO, C. Probability distribution of solution time in grasp: an experimental investigation. *Journal of Heuristics*, v. 8, p. 343–373, 2002. Citado na página 98.
- ALEX, R. M.; RESENDE, M. G. C.; RIBEIRO, C. C. TTT plots: a perl program to create time-to-target plots. *Optimization Letters*, v. 1, p. 355–366, 2007. Citado 2 vezes nas páginas 48 e 98.
- ALUMUR, S.; KARA, B. Y. Network hub location problems: The state of the art. *European Journal of Operational Research*, v. 190, p. 1–21, 2008. Citado 3 vezes nas páginas 22, 58 e 62.
- ALUMUR, S.; NICKEL, S.; GAMA, F. Hub location under uncertainty. *Transportation Research Part B*, v. 46, p. 529–543, 2012. Citado na página 61.
- ALZAMORA, G. S. et al. The air transportation hub-and-spoke design problem: comparison between a continuous and discrete solution method. *Pesquisa Operacional*, v. 33, p. 478–489, 2013. Citado na página 64.
- APPLEGATE, D. et al. *Concorde TSP Solver*. [S.l.], 1996. Citado na página 139.
- ARROYO, J. E. C.; MARQUES T. B. CORTES, J. M. R. Um algoritmo genético para o problema de alocação de antes de transmissão. *XXVI Encontro Nacional de Engenharia de Produção*, 2006. Citado na página 54.

- AYKIN, T. Lagrangian relaxation based approaches to capacitated hub-and-spoke network design problem. *European Journal of Operational Research*, v. 79, p. 501–523, 1994. Citado na página 109.
- AYKIN, T. Network policies for hub-and-spoke systems with applications to the air transportation system. *Transportation Science*, v. 29, p. 201–221, 1995. Citado na página 57.
- AZIZI, N. et al. The impact of hub failure in hub-and-spoke networks: Mathematical formulations and solution techniques. *Computers & Operations Research*, v. 65, p. 174 – 188, 2016. ISSN 0305-0548. Citado na página 61.
- BAEVRE, O. B.; GJENGSTO, B. *Solving combined transportation and production problems using metaheuristic parallelization*. Dissertação (Mestrado) — Molde University College, 2009. Citado na página 24.
- BAILEY, A.; ORNBUKI-BERRNAN, B.; ASOBIELA, S. Discrete pso for the uncapacitated single allocation hub location problem. *Computational Intelligence in Production and Logistics Systems*, v. 1, p. 92–98, 2013. Citado na página 78.
- BALDACCI, R.; AMICO, M. D.; GONZALEZ, J. S. The capacitated m-ring star problem. *Operations Research*, v. 55, p. 1147–1162, 2007. Citado na página 138.
- BELDING, T. The distributed genetic algorithm revisited. *International Conference on Genetic Algorithm*, 1995. Citado na página 38.
- BENDERS, J. F. Partitioning procedures for solving mixed-variables programming problems. *Numerisch Mathematik*, v. 4, p. 238–252, 1962. Citado 2 vezes nas páginas 31 e 62.
- BIANCHI, L. et al. A survey on metaheuristic for stochastic combinatorial optimization. *Journal Natural Computing*, v. 8, p. 239–287, 2009. Citado na página 34.
- BOLAND, N. et al. Preprocessing and cutting for multiple allocation hub location problem. *European Journal of Operational Research*, v. 155, p. 638–653, 2004. Citado 3 vezes nas páginas 109, 110 e 111.
- BOUKANI, F. H.; MOGHADDAM, B. F.; PISHVAEE, M. S. Robust optimization approach to capacitated single and multiple allocation hub location problems. *Computational and Applied Mathematics*, 2016. Citado na página 61.
- BOUSSAID, I.; LEPAGNOT, J.; SIARRY, P. A survey on optimization metaheuristics. *Information Science*, v. 10, p. 82–117, 2013. Citado na página 34.
- BRITO, R. F.; GOMES, B. N.; CAMARGO, R. S. Uma eficiente heurística para o projeto de redes eixo-raio: um estudo de caso para as cidades de minas gerais. *XLVII SBPO, Símposio Brasileiro de Pesquisa Operacional, Porto de Galinhas*, p. 2739–2750, 2015. Citado na página 79.
- BRONDANI, A. E. et al. Aplicação de um modelo de p-medianas para alocação de unidades urbanas de lazer. *SBPO - Símposio Brasileiro de Pesquisa Operacional*, 2012. Citado na página 54.

BRYAN, D. L.; O'KELLY, M. E. Hub-and-spoke networks in air transportation: an analytical review. *Journal of Regional Science*, v. 2, n. 30, p. 275–295, 1999. Citado na página 64.

CALIK, H. et al. A tabu search based heuristic for the hub covering problem over incomplete hub networks. *Computers & Operations Research*, v. 36, n. 12, p. 3088 – 3096, 2009. ISSN 0305-0548. New developments on hub location. Citado na página 64.

CAMARGO, R. S. *Sistemas eixo-raio de multiple alocação: modelos e algoritmos*. Tese (Doutorado) — Universidade Federal de Minas Gerais. Programa de Pós Graduação em Ciência da Computação, 2007. Citado 7 vezes nas páginas 13, 20, 21, 56, 58, 69 e 70.

CAMARGO, R. S.; G., M. J.; FERREIRA, R. P. M. A hybrid outer-approximation/benders decomposition algorithm for the single allocation hub location problem under congestion. *Operations Research Letters*, v. 39, p. 328–337, 2011. Citado na página 62.

CAMARGO, R. S.; MIRANDA, G. Single allocation hub location problem under congestion: Network owner and user perspectives. *Expert Systems with Applications*, v. 39, p. 3386–3391, 2012. Citado na página 65.

CAMARGO, R. S.; Miranda Jr, G.; LUNA, H. P. Benders decomposition for the uncapacitated multiple allocation hub location problem. *Computers and Operations Research*, v. 35, p. 1047–1064, 2008. Citado 4 vezes nas páginas 51, 62, 111 e 128.

CAMARGO, R. S. de; MIRANDA, G. J.; LØKKETANGEN, A. A new formulation and an exact approach for the many-to-many hub location-routing problem. *Applied Mathematical Modelling*, v. 37, n. 12–13, p. 7465 – 7480, 2013. ISSN 0307-904X. Citado na página 62.

CAMPBELL, J. A survey of network hub location. *Studies in Locational Analysis*, v. 6, p. 31–49, 1994. Citado na página 22.

CAMPBELL, J. F. Integer programming formulations of discrete hub location problems. *European Journal of Operational Research*, v. 72, p. 387–405, 1994. Citado 3 vezes nas páginas 22, 57 e 109.

CAMPBELL, J. F. Integer programming formulations of discrete hub location problems. *European Journal of Operational Research*, v. 72, p. 387–405, 1994. Citado na página 74.

CAMPBELL, J. F.; ERNST, A. T.; KRISHNAMOORTHY, M. Hub location problems. In: DREZNER, Z.; HAMACHER, H. W. (Ed.). *Facility Location: Applications and Theory*. 1ª. [S.l.]: Springer, 2002. cap. 12, p. 373–407. Citado na página 22.

CAMPBELL, J. F.; O'KELLY, M. E. Twenty-five years of hub location research. *Transportation*, v. 46, p. 153–169, 2012. Citado na página 62.

CANOVAS, L.; GARCIA, S.; MARIN, A. Solving the uncapacitated multiple allocation hub location problem by means of a dual-ascent technique. *Departamento de Estadística e Investigación Operativa, University of Murcia*, 2004. Citado 2 vezes nas páginas 110 e 111.

CARELLO, G. et al. Solving the hub location problem in telecommunication network design: A local search approach. *Networks*, v. 44, p. 94–105, 2004. Citado na página 70.

- CARVALHO, R. et al. A parallel heuristics for the single allocation hub location problem. *IEEE Latin American*, v. 15, p. 1278–1285, 2017. Citado na página 97.
- CASTRO, R.; CAMARGO, R. S.; MIRANDA, G. Método de decomposição de benders aplicado a localização de concentradores em redes do tipo eixo-raio com alocação simples. *SBPO, Símpoio Brasileiro de Pesquisa Operacional*, 2009. Citado na página 76.
- CASTRO, R. R. M. de. *Sistemas Eixo-Raio de Alocação Simples: Modelos e Algoritmos*. Dissertação (Mestrado) — Department of Industrial Engineering, Federal University of Minas Gerais, December 2010. Citado na página 76.
- CHANDRA, R. et al. *Parallel Programming in OpenMP*. [S.l.]: Morgan Kaufman, 2001. Citado na página 42.
- CHAPMAN, B.; JOST, G.; PAS, R. B. D. *Using OpenMP: Portable shared memory parallel programming*. [S.l.]: MIT, 2008. Citado na página 42.
- CHEN, J. A hybrid heuristic for the uncapacitated single allocation hub location problem. *Omeqa*, v. 35, p. 211–220, 2007. Citado 4 vezes nas páginas 63, 76, 79 e 127.
- CHEN, J. F. A heuristic for the uncapacitated multiple allocation hub location problem. *Jornal of the Chinese Institute of Industrial Engineers*, v. 23, p. 371–381, 2006. Citado 2 vezes nas páginas 110 e 112.
- CHEN, W.; HE, K.; FANG, X. Optimization of hybrid hub-and-spoke network operation for less-than-truckload freight transportation considering incremental quantity discount. *Mathematical Problems in Engineering*, 2014. Citado na página 71.
- CÁNOVAS, L.; GARCIA, S.; MARIN, A. Solving the uncapacitated multiple allocation hub location problem by means a dual ascent technique. *European Journal of Operation Research*, 2007. Citado 2 vezes nas páginas 62 e 111.
- COHOON, J. et al. Punctuated equilibria: A parallel genetic algorithm. In: *2nd International Conference on Genetic Algorithm*. [S.l.: s.n.], 1987. Citado na página 38.
- CONTRERAS, I.; CORDEAU, J. F.; LAPORTE, G. Benders decomposition for large-scale uncapacitated hub location. *Operations Research*, 2011. Citado 2 vezes nas páginas 62 e 111.
- CONTRERAS, I.; FERNÁNDEZ, E. General network design: A unified view of combined location and network design problems. *European Journal of Operation Research*, v. 219, n. 3, p. 680–697, 2012. Citado na página 137.
- CONTRERAS, I.; FERNÁNDEZ, E.; MARÍN, A. Tight bounds from a path based formulation for the tree of hub location problem. *Comput. Oper. Res.*, v. 36, n. 12, p. 3117–3127, 2009. Citado na página 60.
- CONTRERAS, I.; FERNÁNDEZ, E.; MARÍN, A. The tree of hubs location problem. *European Journal of Operational Research*, v. 202, p. 390–400, 2010. Citado 2 vezes nas páginas 60 e 67.
- CONTRERAS, I.; TANASH, M.; VIDYARTHI, N. Exact and heuristic approaches for the cycle hub location problem. *Annals of Operations Research*, p. 1–23, 2016. Citado 5 vezes nas páginas 63, 68, 135, 138 e 143.

- COOK, G. N. Airline networks: A comparison of hub-and-spoke and point-to-point systemairline networks: A comparison of hub-and-spoke and point-to-point systems. *Journal of Aviation/Aerospace Education and Research*, v. 17, p. 51–60, 2008. Citado na página 66.
- COOK, R. P. *An Introduction to Parallel Programming with OpenMP, PThreads and MPI*. [S.l.: s.n.], 2011. Citado na página 42.
- CORBERÁN, A. et al. Strategic oscillation for the capacitated hub location problem with modular links. *Journal of Heuristics*, v. 22, p. 221–244, 2016. Citado na página 64.
- CORMEN, T. H. et al. *Introduction to Algorithms*. [S.l.]: MIT Press, 1990. Citado na página 23.
- CORREA, E. S. *Algoritmos genéticos e busca tabu aplicados ao problema das p-medians*. Dissertação (Mestrado) — Universidade Federal do Paraná, 2000. Citado na página 54.
- COSTA, M. G.; CAPTIVO, M. E.; CLIMACO, J. Capacitated single allocation hub location problem. a bi-criteria approach. *Computers and Operations Research*, v. 35, p. 3671–3695, 2008. Citado na página 61.
- CRAINIC, T. G. *Parallel Metaheuristic*. Interuniversity Research entre on Enterprise Networks, Logistics and Transportation (CIRRELT), 2015. Citado na página 42.
- CRAINIC, T. G.; TOULOUSE, M. *Parallel Meta-Heuristics*. Interuniversity Research entre on Enterprise Networks, Logistics and Transportation (CIRRELT), 2009. Citado 2 vezes nas páginas 35 e 36.
- CUNG, V.-D. et al. Strategies for the parallel implementation of metaheuristic. *Essays and Surveys in Metaheuristics*, 2002. Citado na página 36.
- CUNHA, C.; SILVA, M. A genetic algorithm for the problem of configuring a hub-and-spoke network for a LTL trucking company in Brazil. *European Journal of Operational Research*, v. 179, p. 747–758, 2007. Citado 5 vezes nas páginas 21, 63, 71, 75 e 79.
- CURRENT, J. R.; SCHILLING, D. A. The median tour and maximal covering tour problems: Formulations and heuristics. *European Journal of Operational Research*, v. 73, p. 114–126, 1994. Citado na página 137.
- DAGANZO, C. F. *Logist System Analysis*. [S.l.]: Springer, 2005. Citado na página 64.
- DJUKIC, M. *A hybrid genetic algorithm for solving the uncapacitated multiple allocation hub location problem*. Dissertação (Mestrado) — Univesity of Belgrade, Faculty of Mathematics, 2010. Citado 2 vezes nas páginas 110 e 127.
- DOBSON, G.; LEDERER, P. Airline scheduling and routing in a hub-and-spoke system. *Tranportation Science*, p. 281–297, 1993. Citado na página 65.
- DREZNER, Z.; HAMACHER, H. *Facility location*. [S.l.]: Springer-Verlag, 1995. Citado na página 54.
- DUECK, G.; SCHEUER, T. Threshold accepting: a general purpose optimization algorithm appearing superior to simulated annealing. *Journal of computational physics*, v. 90, p. 161–175, 1990. Citado na página 78.

- DUHAMEL, C. et al. Hybrid metaheuristics. In: \_\_\_\_\_. [S.l.]: Springer, 2013. cap. A Multi-thread GRASPxEELS for the heterogeneous Capacitated Vehicle Routing Problem, p. 237–269. Citado na página 37.
- DURAN, M.; GROSSMANN, I. E. An outer-approximation algorithm for a class of mixed integer nonlinear programs. *Mathematical Programming*, v. 36, p. 307–339, 1986. Citado na página 62.
- EBERY, J. Solving large single allocation p-hub problems with two or three hubs. *European Journal of Operational Research*, v. 128, n. 2, p. 447–458, 2001. Citado na página 57.
- EBERY, J. et al. The capacitated multiple allocation hub location problema: Formulations and algorithms. *European Journal of Operational Research*, v. 120, p. 614–631, 2000. Citado na página 51.
- ELHEDHLI, S.; HU, F. Hub-and-spoke network design with congestion. *Computers and Operations Research*, v. 32, p. 1615–1632, 2005. Citado na página 65.
- ERNST, A. T.; KRISHNAMOORTHY, M. Efficient algorithms for the uncapacitated single allocation p-hub median problem. *Location Science*, v. 4, p. 139–154, 1996. Citado na página 57.
- ERNST, A. T.; KRISHNAMOORTHY, M. Exact and heuristic algorithms for the uncapacitated multiple allocation p-hub median problem. *European Journal of Operational Research*, v. 104, p. 100–112, 1998. Citado na página 109.
- ERNST, A. T.; KRISHNAMOORTHY, M. An exact solution approach based on shortest-paths for p-hub median problems. *INFORMS Journal on Computing*, INFORMS, v. 10, n. 2, p. 149–162, 1998. Citado na página 114.
- ERNST, A. T.; KRISHNAMOORTHY, M. An exact solution approach based on shortest-paths for p-hub median problems. *Journal on Computing*, v. 10, p. 149–162, 1998. Citado na página 110.
- FARAHANI, R. Z. et al. Hub location problems: A review of models, classification, solution techniques, and applications. *Computers & Industrial Engineering*, v. 64, p. 1096–1109, 2013. Citado 4 vezes nas páginas 57, 58, 62 e 66.
- FILIPOVIC, V. An eletromagnetism metaheuristic for the uncapacitated multiple allocation hub location problem. *Journal of Computing*, v. 5, p. 261–272, 2011. Citado na página 112.
- FILIPOVIC, V. et al. Ga inspired heuristic for uncapacitated single allocation hub location problem. *Applications of soft Computing*, v. 58, p. 149–158, 2009. Citado na página 63.
- FLOYD, R. W. Algorithm 97: Shortest path. *Commun. ACM*, ACM, New York, NY, USA, v. 5, n. 6, p. 345–, jun. 1962. ISSN 0001-0782. Disponível em: <<http://doi.acm.org/10.1145/367766.368168>>. Citado na página 113.
- FLYNN, M. J. Some computer organizations and their effectiveness. *IEEE Transactions on Computers*, v. 21, p. 948–960, 1972. Citado na página 39.

FRIEDMAN, M. The use of ranks to avoid the assumption of normality in the analysis of variance. *Journal of the American Statistical Association*, v. 32, n. 200, p. 675–701, 1937. Citado 2 vezes nas páginas 50 e 99.

GARCIA, V. J. et al. Grasp para o problema de planejamento de redes secundárias de distribuição de energia elétrica. *XXXV SBPO - Simpósio Brasileiro de Pesquisa Operacional*, 2003. Citado na página 54.

GAVISH, B. Optimization models for configuring distributed computer systems. *IEEE Transactions on Computers*, C-36, n. 7, p. 773–793, 1987. Citado na página 68.

GELAREH, S. *Hub Location Models in Public Transport Planning*. Tese (Doutorado) — University of Saarlandes, 2008. Citado na página 68.

GENDREAU M. AND POTVIN, J. *Handbook of Metaheuristic*. [S.l.]: Springer, 2010. v. 146. (International Series in Operations Research and Management Science, v. 146). Citado na página 34.

GENOVA, K.; GULIASHKI, V. Linear integer ‘programming methods and approaches. *Cybernetics and Information Technologies*, 2011. Citado na página 31.

GERLACH, S. Hub location models in public transport planning. *Tese de Doutorado apresentada a Universidade de Sarrlandes, Alemanha.*, 2008. Citado na página 60.

GOLDMAN, A. J. Optimal locations for centers in a network. *Transportation Science*, p. 352–360, 1969. Citado na página 57.

GOMES, B. N. Projeto eixo-raio com alocação simples: Modelos e métodos. In: *Tese apresentada ao Programa de Pós Graduação em Engenharia Elétrica, UFMG*. [S.l.: s.n.], 2016. Citado na página 97.

GOMES, B. N. et al. An efficient genetic algorithm for the design of hub-and-spoke networks. *IEEE Communications Letters*, 2013. Citado 4 vezes nas páginas 63, 78, 79 e 82.

GONCALVES, A. A. C.; SOUZA, S. R.; SILVA, C. A. Uma metodologia de hibridização heurística para resolução do problema de roteamento de veículos com coleta e entrega e janela de tempo. In: *XLVIII SBPO, Simposio de Pesquisa Operacional*. [S.l.: s.n.], 2016. Citado na página 44.

GRAMA, A. et al. *Introduction to Parallel Computing*. [S.l.]: Addison Wesley, 2003. Citado na página 89.

GRAMA, A.; KUMAR, V. State of the art in parallel search techniques for discrete optimization problems. *IEEE Transactions on Knowledge and Data Engineering*, v. 11, p. 28–35, 1999. Citado na página 36.

GRENEAU, M.; LAPORTE, G.; SEMET, F. The covering tour problem. *Operational Research*, v. 45, n. 5, p. 568–576, 1997. Citado na página 137.

HAKIMI, S. L. Optimum distribution of switching centres in a communication network and some related graph theoretic problems. *Operations Research*, v. 13, p. 462–475, 1965. Citado na página 54.

- HALL, R. W. Configuration of an overnight package air network. *Transportation Research*, v. 23, p. 139–149, 1989. Citado na página 65.
- HAMACHER, H. W. et al. *Polyhedral properties of the uncapacitated multiple allocation hub location problem*. <http://www.itwm.fhg.de/zentral/berichte/bericht20.pdf>, 2000. Citado 3 vezes nas páginas 108, 109 e 111.
- HOFFMAN, K. L.; PADBERG, M. Solution techniques for integer programming. *Combinatorial and Integer Optimization*, 1999. Citado na página 31.
- HOSAPUJARI, A. B.; VERMA, A. Development of a hub and spoke model for bus transit route network design. *Procedia - Social and Behavioral Sciences*, v. 104, p. 835–844, 2013. Citado na página 67.
- ISHFAQ, R.; SOX, C. R. Hub location–allocation in intermodal logistic networks. *European Journal of Operational Research*, v. 210, n. 2, p. 213 – 230, 2011. ISSN 0377-2217. Citado na página 64.
- ISHFAQ, R.; SOX, C. R. Design of intermodal logistics networks with hub delays. *European Journal of Operational Research*, v. 220, n. 3, p. 629 – 641, 2012. ISSN 0377-2217. Citado na página 64.
- JAILLET, P.; SONG, G.; YU, G. Airline network design and hub location problems. *Location Science*, v. 4, p. 195–211, 1996. Citado na página 65.
- JUNGUER, M.; REINELT, G.; THIENEL, S. Practical problem solving with cutting plane algorithms in combinatorial optimization. *Dynamics Series in Discrete Mathematics and Theoretical Computer Science*, v. 20, p. 111–152, 1995. Citado na página 31.
- KARA, B. Y.; TANSEL, B. C. On the single-assignment p-hub center problem. *European Journal of Operational Research*, v. 125, p. 648–655, 2000. Citado na página 65.
- KIM, H.; O’KELLY, M. Reliable p-hub location problems in telecommunication networks. *Geographical Analysis*, v. 41, p. 283–306, 2009. Citado na página 70.
- KIM, M. J.; KIM, S. B.; RYU, K. H. The capacitated hierarchical p-median problem for facility location of pcs networks. *Fourth International Conference on Telecommunications System, Nashville.*, 1996. Citado na página 69.
- KLINCEWICZ, J. G. Heuristics for the  $p$ -hub location problem. *European Journal of Operational Research*, v. 53, p. 25–37, 1991. Citado na página 63.
- KLINCEWICZ, J. G. Avoiding local optima in the  $p$ -hub location problem using tabu search and grasp. *Annals of Operations Research*, v. 40, p. 283–302, 1992. Citado na página 63.
- KLINCEWICZ, J. G. A dual algorithm for the uncapacitated hub location problem. *Location Science*, v. 4, n. 3, p. 173–184, 1996. Citado na página 109.
- KLINCEWICZ, J. G. Hub location in backbone/tributary network design: a review. *Location Science*, v. 6, p. 307–335, 1998. Citado na página 68.
- KLINCEWICZ, J. G. Enumeration and search procedures for a hub location problem with economies of scale. *Annals of Operations Research*, v. 110, p. 107–122, 2002. Citado na página 64.

- KRAMER, R. H. F. R.; SUBRAMANIAN, A.; PENNA, P. H. V. Problema de roteamento de veículos assimétrico com frota heterogênea limitada: um estudo de caso em uma indústria de bebidas. *Gestão e Produção*, v. 23, p. 165–176, 2015. Citado na página 44.
- KRATICA, J. et al. Genetic algorithm for solving uncapacitated multiple allocation hub location problem. *Computing and Informatics*, 2005. Citado na página 110.
- KRATICA, J. et al. Two genetic algorithms for solving the uncapacitated single allocation p-hub median problem. *European Journal of Oper*, v. 182, p. 15–28, 2007. Citado 3 vezes nas páginas 63, 127 e 128.
- KUROKI, Y.; MATSUI, T. Approximation algorithm for cycle-star hub network design problems and cycle-metric labeling problems. In: POOH, S. H.; RAHMAN, M. (Ed.). *WALCOM: Algorithms and Computation*. [S.l.: s.n.], 2017. Citado na página 137.
- LABBE, M. et al. The ring star problem: Polyhedral analysis and exact algorithm. *Networks*, v. 43, p. 177–189, 2004. Citado na página 59.
- LABBE, M. et al. Location median cycles in networks. *European Journal of Operational*, 2005. Citado na página 137.
- LABBÉ, M.; YAMAN, H. Projecting flow variables for hub location problems. *Networks*, v. 44, n. 2, p. 84–93, 2004. Citado na página 137.
- LABBÉ, M.; YAMAN, H.; GOURDIN, E. A branch and cut algorithm for hub location problems with single assignment. *Mathematical Programming: Series A*, v. 102, p. 371–405, 2005. Citado na página 62.
- LEE, C. Y. An algorithm for the design of multitype concentrator networks. *Journal of the Operational Research Society*, v. 33, p. 471–482, 1993. Citado na página 69.
- LEE, Y.; CHIU, S.; SANCHEZ, J. A branch and cut algorithm for the steiner ring star problem. *International Journal of Management Science*, p. 21–34, 1998. Citado na página 138.
- LIN, C. C.; LIN, Y. L. The economic effects of center-to-center directs on hub-and-spoke networks for air express common carriers. *Journal of Air Transport Management*, v. 9, p. 255–265, 2003. Citado na página 66.
- LIN, S.; KERNIGHAN, W. An effective heuristic algorithm for the traveling-salesman problem. *Operations Research*, v. 21, p. 498–516, 1973. Citado na página 139.
- LOURENCO, H. R.; MARTIN, O. C.; STUTZLE, T. Iterated local search: Framework and applications. *Handbook of Metaheuristics*, v. 146, p. 363–397, 2010. Citado na página 43.
- LUMSDEN, K.; DALLARI, F.; RUGGERI, R. Improving the efficiency of the hub and spoke system for the skf european distribution network. *International Journal of Physical Distribution & Logistics Management*, v. 19, p. 50–66, 1998. Citado na página 71.
- MARIANOV, V.; SERRA, D. Location models for airline hubs behaving as m/d/c queues. *Computers & Operations Research*, v. 30, n. 7, p. 983 – 1003, 2003. ISSN 0305-0548. Citado 2 vezes nas páginas 61 e 64.

- MARIANOV, V.; SERRA, D. Location models for airline hubs behaving as m/d/c queues. *Computer and Operations Research*, Elsevier Science Ltd., v. 30, n. 7, p. 983–1003, 2003. ISSN 0305-0548. Citado na página 65.
- MARIC, M.; STANIMIROVIC, Z.; STANOJEVIC, P. An efficient memetic algorithm for the uncapacitated single allocation hub location problem. *Soft Comput*, v. 17, p. 445–466, 2013. Citado 2 vezes nas páginas 63 e 78.
- MARÍN, A. Uncapacitated euclidian hub location: Strengthened formulation, new facets and a relax-and-cut algorithm. *Journal of Global Optimization*, v. 33, p. 392–422, 2005. Citado na página 109.
- MARTI, R.; REINELT. *The linear OrdOrder Problem: Exact and Heuristic Methods in Combinatorial Optimization*. [S.l.]: Springer, 2011. Citado 2 vezes nas páginas 32 e 34.
- MARTIN, I. R.; GONZALEZ, J. J.; YAMAN. A branch-and-cut algorithm for the hub location and routing problem. *Computers and Op*, v. 50, p. 161–174, 2015. Citado na página 62.
- MARTINS, A. X.; COSTA, Y. M.; CAMARGO, R. S. Grasp com reconexão por caminhos para o problema eixo-raio com alocação simples. *VI Congresso Latino Iberoamericano de Investigacion Operativa / XLIV Simpósio Brasileiro de Pesquisa Operacional*, 2012. Citado 3 vezes nas páginas 77, 78 e 79.
- MATTSON, T. G.; SANDERS, B. A.; MASSINGILL, B. L. *Patterns for Parallel Programming*. 1st. ed. [S.l.]: Addison-Wesley Professional, 2013. ISBN 0321940784, 9780321940780. Citado na página 42.
- MAYER, G.; WAGNER, B. Hublocator: an exact solution method for the multiple allocation hub location problem. *Computer & Operations Research*, v. 29, p. 715–739, 2002. Citado na página 109.
- MLADENOVIC, N.; HANSEN, P. A variable neighborhood search. *Computers and Operations Research*, v. 24, p. 1097–1100, 1997. Citado 2 vezes nas páginas 43 e 77.
- MONTGOMERY, D. C.; RUNGER, G. C. *Applied Statistics and Probability for Engineers*. [S.l.]: J. W. Sons, 2003. Citado 2 vezes nas páginas 49 e 127.
- NAEEM, M.; BERMAN, B. O. An efficient genetic algorithm for the uncapacitated single allocation hub location problem. *Evolutionary Computation (CEC), 2010 IEEE Congress on*, p. 1–8, 2010. Citado 2 vezes nas páginas 63 e 77.
- NEGREIROS, M.; PALHANO, A. The capacitated centred clustering problem. *Computer and*, 2006. Citado na página 51.
- NEMHAUSER, G.; WOLSEY, A. *Integer and Combinatorial Optimization*. [S.l.]: Wiley and Sons, 1988. Citado na página 31.
- NICHOLS, B.; BUTTLAR, D.; FARRELL, J. *Pthreas programming*. [S.l.]: O’Reilly Media, 1996. Citado na página 42.
- NICKEL, S.; SCHOBEL, A.; SONNEBORN, T. Hub location problems in urban traffic networks. In: NIITTYMAHI; PURSULA (Ed.). *Mathematical Methods and Optimisation in Transportation Systems*. [S.l.]: Kluwer Academic Publisher, 2000. v. 48, p. 1–12. Citado 2 vezes nas páginas 60 e 66.

- O'KELLY, M. et al. Hub location problems with price sensitive demands. *Networks and Spatial Economics*, 2015. Citado na página 61.
- O'KELLY, M. E. Activity levels at hub facilities in interacting networks. *Geographical An*, 1986. Citado na página 57.
- O'KELLY, M. E. The location of interacting hub facilities. *Transportation Science*, v. 20, p. 92–106, 1986. Citado na página 57.
- O'KELLY, M. E. A quadratic integer program for the location of interacting hub facilities. *European Journal of Operational Research*, v. 32, p. 393–404, 1987. Citado 2 vezes nas páginas 57 e 63.
- O'KELLY, M. E. Hub facility location with fixed costs. *Papers in Regional Science*, v. 71, n. 3, p. 293–306, 1992. Citado na página 74.
- O'KELLY, M. E. et al. Hub network design with single and multiple allocation: A computational study. *Location Science*, v. 4, n. 3, p. 125–138, 1996. Citado na página 109.
- O'KELLY, M. E.; LAO, Y. Mode choice in hub-and-spoke network: A zero-one linear programming approach. *Geographical Analysis*, v. 23, p. 283–297, 1991. Citado na página 65.
- O'KELLY, M. E.; MULLER, H. J. The hub network design problem. *Journal of Transport Geography*, v. 2, p. 31–40, 1994. Citado na página 56.
- OKTAL, H.; OZGER, A. Hub location in air cargo transportation: A case study. *Journal of Air Transport Management*, v. 27, p. 1–4, 2013. Citado na página 66.
- OSMAN, I. H. Preface: Focused issue on applied meta-heuristic. *Computer Engineering*, v. 44, p. 205–207, 2003. Citado na página 34.
- OSMAN, I. H.; KELLY, J. P. *Meta-Heuristic: Theory and Applications*. [S.l.]: Kluwer Academic Publisher, 1996. Citado na página 34.
- PAIS, M. S. *Estudo da influencia dos parametros de algoritmos paralelos da computacao evolutiva no seu desempenho em plataformas multicore*. Tese (Doutorado) — Universidade Federal de Uberlandia, 2014. Citado na página 89.
- PAMUK, F. S.; SEPIL, C. A solution to the hub center problem via a single-relocation algorithm with tabu search. *IIE Transactions*, v. 33, p. 399–411, 2001. Citado na página 64.
- PENNA, P. H. V.; SUBRAMANIAN, A.; OCHI, L. S. An iterated local search heuristic for the heterogeneous fleet vehicle routing problem. *Journal of Heuristics*, v. 19, n. 2, p. 201–232, 2013. Citado na página 44.
- PEREIRA, F. E. M. Implementacao de um algoritmo paralelo e escalavel do ant colony system aplicado ao problema do roteamento de veiculos. *Trabalho de conclusao e curso, Instituto Federal do Rio Grande do Norte*, 2015. Citado na página 89.
- PIRKUT, H.; NAGARAJAN, V. Locating cconcentrator in centralized computer network. *Annals of Operations Reseach*, v. 36, p. 247–262, 1992. Citado na página 69.

- PIZZOLATO, N. D.; BARCELOS, A. G. Local de escolas públicas, síntese de algumas linhas de experiência no Brasil. *Pesquisa Operacional*, v. 24, p. 111–131, 2004. Citado na página 54.
- PIZZOLATO, N. D.; RAUPP, G. S. Revisão de desafios aplicados em localização com base em modelos da p-mediana e suas variantes. *Pesquisa Operacional para o Desenvolvimento*, 2012. Citado na página 55.
- PUCHINGER, J.; RAIDL, G. R. Combining metaheuristics and exact algorithms in combinatorial optimization: A survey and classification. In: *International Work-Conference on the Interplay Between Natural and Artificial Computation*. [S.l.: s.n.], 2005. Citado na página 31.
- QU, B.; WENG, K. Path relinking approach for multiple allocation hub maximal covering problem. *Computers & Mathematics with Applications*, v. 57, n. 11–12, p. 1890 – 1894, 2009. ISSN 0898-1221. Proceedings of the International Conference Bio-Inspired Computing-Theories and Applications BIC-TA 2007 Zhengzhou, China. Citado na página 63.
- QUIM, M. J. *Parallel Programming in C with MPI and OpenMP*. [S.l.]: McGraw-Hill Higher Education, 2003. Citado na página 42.
- REAL, L. B. *A Gateway Hub Location Problem*. Dissertação (Mestrado) — Universidade Federal de Minas Gerais, Brasil, 2015. Citado na página 61.
- REESE, J. Methods for solving the p-median problem: An annotated bibliography. *Tech. Rep. Trinity University*, 2005. Citado na página 54.
- ROSÁRIO, R. R. L.; CARNIERI, C.; STEINER, S. A. Aplicação de problema das p-medianas para determinar a localização de unidades de saúde 24 horas. *SBPO - Simpósio Brasileiro de Pesquisa Operacional*, 2002. Citado na página 54.
- RUFINO, A. et al. Metaheurística ILS para o problema de posicionamento automático de pontos de acesso em ambientes internos. In: *XLVIII SBPO, Simpósio Brasileiro de Pesquisa Operacional*. [S.l.: s.n.], 2016. p. 1884–1895. Citado na página 44.
- RUSLIN, N. M.; GHANI, A. An application of the p-median problem with uncertainty in demand in emergency medical services. *Proceedings of the 2nd IMTGT Regional Conference on Mathematics, Statistics and Applications*, 2006. Citado na página 54.
- SÁ, E. M. de. *Design of hub-and-spoke networks applied to public transportation systems*. Tese (Doutorado) — Universidade Federal de Minas Gerais, Programa de Pós-Graduação em Engenharia de Produção, 2015. Citado 2 vezes nas páginas 20 e 66.
- SÁ, E. M. de; CAMARGO, R. S.; MIRANDA, G. Redes eixo-raio aplicada ao transporte público. *XLII SBPO, Simpósio Brasileiro de Pesquisa Operacional*, p. 2415–2426, 2010. Citado na página 60.
- SÁ, E. M. de; CONTRERAS, I.; CORDEAU, J. F. Exact and heuristic algorithms for the design of hub networks with multiple lines. *European Journal of Operational Research*, v. 246, p. 186–198, 2015. Citado na página 68.
- SÁ, E. M. de et al. The hub line location problem. *Transportation Science*, v. 49, p. 500–518, 2013. Citado 2 vezes nas páginas 62 e 67.

- SABERI, M.; MAHMASSANI, H. Modeling the airline hub location and optimal market problems with continuous approximation techniques. *Journal of Transport Geography*, v. 30, p. 68–76, 2013. Citado na página 64.
- SANTOS, V. L.; ARROYO, J. E. C. Sequenciamento de tarefas em máquinas paralelas considerando desgastes dependentes da sequência. *XLVII SBPO, Símposio Brasileiro de Pesquisa Operacional*, p. 2572–2583, 2015. Citado na página 44.
- SASAKI, M.; SUZUKI, A.; DREZNER, Z. On the selection of hub airports for an airline hub-and-spoke. *Computer & Operations Research*, v. 26, p. 1411–1422, 1999. Citado na página 65.
- SHAW S, I. Hub structures of major us passenger airlines. *Journal of Transport Geography*, v. 1, p. 47–58, 1993. Citado na página 65.
- SILVA, M. et al. A simple and effective metaheuristic for the minimum latency problem. *European Journal of Operational Research*, v. 221, n. 3, p. 513 – 520, 2012. ISSN 0377-2217. Citado na página 44.
- SILVA, M. R. *Uma contribuição ao problema de localização de terminais de consolidação no transporte de carga parcelada*. Tese (Doutorado) — Escola Politécnica da Universidade de São Paulo, 2004. Citado 3 vezes nas páginas 21, 22 e 64.
- SILVA, M. R.; CUNHA, C. B. New simple and efficient heuristics for the uncapacitated single allocation hub location problem. *Computers and Operations Research*, v. 36, n. 12, p. 3152–3165, 2009. Citado 3 vezes nas páginas 76, 78 e 79.
- SILVA, V. M. R.; FROTA, Y.; SIMONETTI, L. An ils algorithm for the robust coloring problem. In: *Símposio Brasileiro de Pesquisa Operacional*. [S.l.: s.n.], 2015. Citado na página 44.
- SKORIN-KAPOV, D.; SKORIN-KAPOV, J. On tabu search for the location of interacting hub facilities. *European Journal of Operational Research*, v. 73, p. 501–508, 1994. Citado na página 63.
- SKORIN-KAPOV, D.; SKORIN-KAPOV, J.; O’KELLY, M. Tight linear programming relaxations of uncapacitated  $p$ -hub median problems. *European Journal of Operational Research*, v. 94, p. 582–593, 1996. Citado na página 109.
- SOHN, J.; PARK, S. Efficient solution procedure and reduced size formulations for  $p$ -hub location problems. *European Journal of Operational Research*, v. 108, p. 118–126, 1998. Citado na página 75.
- SOUZA, M. J. F. et al. A hybrid heuristic algorithm for the open-pit-mining operational planning problem. *European Journal of Operational Research*, v. 207, p. 1041–1051, 2010. Citado na página 43.
- STALLINGS, W. *Arquitetura e Organização de Computadores*. [S.l.]: Pearson, 2010. Citado na página 89.
- SUBRAMANIAN, A. et al. A parallel heuristic for the vehicle routing problem with simultaneous pickup and delivery. *Computers and Operations Research*, Elsevier, v. 37, n. 11, p. 1899–1911, 2010. Citado na página 44.

- TAKANO, K.; ARAI, M. A genetic algorithm for the hub-and-spoke problem applied to containerized cargo transport. *Journal of Marine Science and Tec*, v. 14, p. 256–274, 2008. Citado na página 63.
- TALBI, E. *MetaHeuristic from Design to implementation*. [S.l.]: John Wiley and Sons, 2009. Citado 7 vezes nas páginas 13, 34, 35, 36, 37, 38 e 39.
- TANENBAUM, A. S. *Sistemas Operacionais Modernos*. [S.l.]: Pearson, 2008. Citado 2 vezes nas páginas 13 e 41.
- TAYLOR, D. et al. Development and analysis of alternative dispatching methods in truckload trucking. *Transportation Research Part E: Logistics and Transportation Review*, v. 35, p. 191–205, 1999. Citado 2 vezes nas páginas 21 e 71.
- TEYMOURIAN, E.; SADEGHI, A.; TAGHIPOURIAN, F. A dynamic virtual hub location problem in airline networks - formulation and metaheuristic solution approaches. In: *First International Technology Management Conference*. [S.l.: s.n.], 2011. p. 1061–1068. Citado na página 61.
- THOMADSEN, T.; LARSEN, J. A hub location problem with fully interconnected backbone and access networks. *Computers & Operations Research*, v. 34, n. 8, p. 2520 – 2531, 2007. ISSN 0305-0548. Citado na página 62.
- TING, C. J.; WANG, H. J. A threshold accepting algorithm for the uncapacitated single allocation hub location problem. *Journal of Chinese Institute of Engineers*, v. 37, p. 300–312, 2013. Citado na página 78.
- TOH, R. S.; HIGGINS, R. G. The impact of hub-and-spoke network centralization and route monopoly on domestic airline profitability. *Transportation Journal*, v. 24, p. 16–27, 1985. Citado 2 vezes nas páginas 21 e 64.
- TOPCUOGLU, H. et al. Solving the uncapacitated hub location problem using genetic algorithms. *Computers and Operations Research*, v. 32, n. 4, p. 967–984, 2005. Citado 4 vezes nas páginas 63, 75, 76 e 79.
- VERHOEVEN, M.; AARTS, E. H. L. Parallel local search. *Journal of Heuristics*, 1995. Citado na página 36.
- WOLSEY, L. A. *Integer Programming*. [S.l.]: Wiley, 1998. (Wiley Series in Discrete Mathematics and Optimization). Citado 2 vezes nas páginas 30 e 31.
- YAMAN, H. Star p-hub median problem with modwith are capacities. *Computers and Operations Research*, v. 35, p. 3009–3019, 2008. Citado na página 59.
- YAMAN, H.; CARELLO, G. Solving the hub location problem with modular link capacities. *Computers and Operations Research*, v. 32, p. 3227–3245, 2005. Citado na página 64.
- YANG, T. H. Stochastic air freight hub location and flight routes planning. *Applied Mathematical Modelling*, v. 12, p. 4424–4430, 2009. Citado na página 61.
- YOUNG, R. D. A primal (all integer), integer programming algorithm. *Journal of Research of the National Bureau of Standards*, 1965. Citado na página 31.

YU, B. et al. An optimization method for planning the lines and the operational strategies of waterbuses: the case of zhoushan city. *Operational Research*, v. 15, p. 25–49, 2015. Citado na página 67.

YU, B. et al. Two-phase optimization approach to transit hub location - the case of dalian. *Journal of Transport Geography*, v. 33, p. 62–71, 2013. Citado na página 67.

ZAPFEL, G.; WASNER, M. Planning and optimization of hub-and-spoke transportation networks of cooperative third-party logistics providers. *International Journal of Production Economics*, v. 78, p. 207–220, 2002. Citado na página 71.

ZHALECHIAN, M. et al. An interactive possibilistic programming approach for a multi-objective hub location problem: Economic and environmental design. *Applied Soft Computing*, v. 52, p. 699–713, 2017. Citado na página 61.