



UNIVERSIDADE FEDERAL RURAL DO SEMI-ÁRIDO
UNIVERSIDADE DO ESTADO DO RIO GRANDE DO NORTE
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA
COMPUTAÇÃO



IRAMAR FERREIRA DOS SANTOS

MODELAGEM MATEMÁTICA E APLICAÇÃO DA
METAHEURÍSTICA VNS PARA O PROBLEMA DE
PROJETO DE REDE

Mossoró-RN

2020

IRAMAR FERREIRA DOS SANTOS

**MODELAGEM MATEMÁTICA E APLICAÇÃO DA
METAHEURÍSTICA VNS PARA O PROBLEMA DE
PROJETO DE REDE**

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação - associação ampla entre a Universidade do Estado do Rio Grande do Norte e a Universidade Federal Rural do Semi-Árido, para a obtenção do título de Mestre em Ciência da Computação.

Linha de Pesquisa: Otimização e Inteligência Computacional

Orientador: Prof^o Dr. Carlos Heitor Pereira Liberalino,

Coorientador: Prof^o Dr. Francisco Chagas de Lima Júnior

Mossoró-RN

2020

© Todos os direitos estão reservados a Universidade do Estado do Rio Grande do Norte. O conteúdo desta obra é de inteira responsabilidade do(a) autor(a), sendo o mesmo, passível de sanções administrativas ou penais, caso sejam infringidas as leis que regulamentam a Propriedade Intelectual, respectivamente, Patentes: Lei nº 9.279/1996 e Direitos Autorais: Lei nº 9.610/1998. A mesma poderá servir de base literária para novas pesquisas, desde que a obra e seu(a) respectivo(a) autor(a) sejam devidamente citados e mencionados os seus créditos bibliográficos.

Catálogo da Publicação na Fonte.
Universidade do Estado do Rio Grande do Norte.

S237m Santos, Iramar Ferreira dos
Modelagem Matemática e Aplicação da Metaheurística
VNS para o Problema de Projeto de Rede. / Iramar
Ferreira dos Santos. - Mossoró/RN, 2020.
85p.

Orientador(a): Prof. Dr. Carlos Heitor Pereira
Liberalino.

Coorientador(a): Prof. Dr. Francisco Chagas de Lima
Júnior.

Dissertação (Mestrado em Programa de Pós-
Graduação em Ciência da Computação). Universidade do
Estado do Rio Grande do Norte.

1. Problema de Projeto de Rede. 2. Otimização
Combinatória. 3. Metaheurística. I. Liberalino, Carlos
Heitor Pereira. II. Universidade do Estado do Rio Grande
do Norte. III. Título.

IRAMAR FERREIRA DOS SANTOS

**MODELAGEM MATEMÁTICA E APLICAÇÃO DA
METAHEURÍSTICA VNS PARA O PROBLEMA DE
PROJETO DE REDE**

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação para a obtenção do título de Mestre em Ciência da Computação.

APROVADA EM: ____ / ____ / _____.

BANCA EXAMINADORA

Carlos Heitor Pereira Liberalino, Dr.
Orientador e Presidente

Francisco Chagas de Lima Júnior, Dr.
Coorientador - Universidade do Estado do
Rio Grande do Norte

Dario José Aloise, Dr.
Membro Interno - Universidade do Estado
do Rio Grande do Norte

André Pedro Fernandes Neto, Dr.
Membro Externo - Universidade Federal
Rural do Semi-Árido

Dedico este trabalho especialmente a Deus, aos meus pais, Sebastião Fortunado dos Santos e Ivanete Maria Ferreira dos Santos, e aos meus irmãos, Iritan Ferreira dos Santos, Iran Ferreira dos Santos e Ana Cecília Ferreira dos Santos.

Agradecimentos

Primeiramente a Deus que permitiu que tudo isso acontecesse, por sempre ter me dado força, saúde e garra para enfrentar as dificuldades existentes.

À minha família, pelo amor, incentivo, apoio incondicional e por sempre acreditarem em mim, minha mãe Ivanete Maria Ferreira dos Santos, meu pai Sebastião Fortunado dos Santos, meus irmãos Iritan Ferreira dos Santos, Iran Ferreira dos Santos e Ana Cecília Ferreira dos Santos.

Em especial, agradeço à minha namorada Maria das Graças pela paciência, companheirismo, afeto e compreensão nos momentos tristes e difíceis. Obrigado por sempre estar presente durante toda essa caminhada, te amo.

Aos meus colegas, Thiago Pereira e Jenilson Fernandes, sou grato a vocês pelo convívio que tivemos, e por serem minha família na cidade de Mossoró/RN.

Agradeço também aos meus colegas do mestrado, pela troca de conhecimentos e por me ajudarem de uma forma ou de outra durante todo o período do mestrado.

Aos meus orientadores, Carlos Heitor e Francisco Chagas, pela oportunidade de me orientar no desenvolvimento deste trabalho, e pelos ensinamentos e conhecimentos transmitidos durante a pesquisa.

Agradeço a todos os professores por proporcionar valiosos conhecimentos, e por ajudarem a me tornar um mestre com boa bagagem.

À CAPES, pelo apoio financeiro e a UERN e UFERSA pela estrutura que possibilitou o desenvolvimento desta pesquisa.

Resumo

Este trabalho apresenta um problema de projeto de rede (*Network Design Problem - NDP*), o qual é um problema de otimização combinatória classificado como NP-Difícil. O NDP consiste em encontrar um projeto, dentro de vários projetos de modificações da infraestrutura de uma malha viária que promova uma melhoria do seu fluxo total. Neste contexto, propõe-se resolver o NDP com a determinação de vias de mão única e mão dupla em uma malha viária, com o propósito de encontrar uma melhor configuração da malha viária de modo que minimize o tempo total de viagem. É apresentada uma nova formulação matemática para este problema, e o mesmo foi implementado em um *solver* matemático. Com os resultados obtidos do método exato para uma instância neste trabalho, o novo modelo matemático proposto é válido. Ainda faz parte deste trabalho, um estudo algorítmico através da implementação, experimentos computacionais e análises dos resultados para o método de solução desenvolvido para o NDP. O método de solução desenvolvido consiste na utilização do algoritmo de Floyd-Warshall para encontrar os caminho mais curtos para todos os viajantes na malha viária entre origens e destinos, e na aplicação da metaheurística VNS (*Variable Neighborhood Search*) ao NDP. Foram criadas novas instâncias e também utilizou-se instâncias já conhecida na literatura para os experimentos computacionais. Com os resultados obtidos, a metaheurística VNS mostrou ser promissora em alcançar soluções boas em um tempo viável, ao invés de utilizar a abordagem exata, e também, conseguiu resultados melhores do que os já conhecidos na literatura.

Palavras-chave: Problema de Projeto de Rede, Otimização Combinatória, Metaheurística.

Abstract

This work presents a network design problem - NDP, which is a combinatorial optimization problem classified as NP-Hard. The NDP consists of finding a project within several projects of modifications of the infrastructure of a road network that promotes an improvement of its total flow. In this context, it is proposed to solve the NDP by determining one-way and two-way lanes in a road network, in order to find a better road network configuration so as to minimize the total travel time. It is presented a new mathematical formulation for this problem, and it was implemented in a mathematical solver. With the results obtained from the exact method for an instance in this work, the proposed new mathematical model is valid. Still part of this work, an algorithmic study through the implementation, computational experiments and analysis of results for the solution method developed for NDP. The solution method developed consists of using the Floyd-Warshall algorithm to find the shortest paths for all travelers in the origin-destination road network, and in the application of the VNS (Variable Neighborhood Search) metaheuristic to the NDP. New instances were created and instances already known in the literature were used for computational experiments. With the results obtained, the VNS metaheuristic proved to be promising to reach good solutions in a viable time, instead of using the exact approach, and also, obtained better results than already known in the literature.

Keywords: Network Design Problem, Combinatorial Optimization, Metaheuristic.

Lista de ilustrações

Figura 1 – Ilustração de soluções em problemas de otimização combinatória.	23
Figura 2 – Ilustração do procedimento de execução do algoritmo de Floyd-Warshall.	27
Figura 3 – Ilustração do NDP como um problema de dois níveis.	32
Figura 4 – Fluxograma do método de solução proposta.	38
Figura 5 – Ilustração da malha viária e o grafo que a representa.	39
Figura 6 – Representações de uma solução.	41
Figura 7 – Comparação dos resultados obtidos para as três configurações dos experimentos no CPLEX para a instância PQ_14_20.	50
Figura 8 – Comparação do tempo de execução para as três configurações dos experimentos no CPLEX para a instância PQ_14_20.	51
Figura 9 – Comparação das soluções do método exato e do VNS para a instância PQ_18_41.	54
Figura 10 – Comparação das soluções do método exato e do VNS para a instância SF_21_38.	55
Figura 11 – Comparação das melhores soluções dos algoritmos da literatura e do VNS para a instância MD_40_164.	56
Figura 12 – Comparação das médias das soluções dos algoritmos da literatura e do VNS para a instância MS_40_65.	57
Figura 13 – Comparação das médias das soluções dos algoritmos da literatura e do VNS para a instância ME_40_99.	58
Figura 14 – Comparação das médias das soluções dos algoritmos da literatura e do VNS para a instância MD_40_164.	58
Figura 15 – Grafo da Solução inicial para a instância PQ_14_20.	73
Figura 16 – Grafo da melhor solução encontrada para a instância PQ_14_20 com $\alpha = 0.8$	73
Figura 17 – Grafo da melhor solução encontrada para a instância PQ_14_20 com $\alpha = 0.7$	74
Figura 18 – Grafo da melhor solução encontrada para a instância PQ_14_20 com $\alpha = 0.6$	74
Figura 19 – Grafo da melhor solução encontrada para a instância PQ_14_20 com $\alpha = 0.5$	74
Figura 20 – Grafo da Solução inicial para a instância PQ_18_41.	75

Figura 21 – Grafo da melhor solução encontrada para a instância PQ_18_41 com $\alpha = 0.8$	75
Figura 22 – Grafo da melhor solução encontrada para a instância PQ_18_41 com $\alpha = 0.7$	75
Figura 23 – Grafo da melhor solução encontrada para a instância PQ_18_41 com $\alpha = 0.6$	76
Figura 24 – Grafo da melhor solução encontrada para a instância PQ_18_41 com $\alpha = 0.5$	76
Figura 25 – Grafo da Solução inicial para a instância SF_21_38.	76
Figura 26 – Grafo da melhor solução encontrada para a instância SF_21_38 com $\alpha = 0.8$	77
Figura 27 – Grafo da melhor solução encontrada para a instância SF_21_38 com $\alpha = 0.7$	77
Figura 28 – Grafo da melhor solução encontrada para a instância SF_21_38 com $\alpha = 0.6$	77
Figura 29 – Grafo da melhor solução encontrada para a instância SF_21_38 com $\alpha = 0.5$	78
Figura 30 – Grafo da Solução inicial para a instância MS_40_65.	78
Figura 31 – Grafo da melhor solução encontrada para a instância MS_40_65 com $\alpha = 0.8$	79
Figura 32 – Grafo da melhor solução encontrada para a instância MS_40_65 com $\alpha = 0.7$	79
Figura 33 – Grafo da melhor solução encontrada para a instância MS_40_65 com $\alpha = 0.6$	80
Figura 34 – Grafo da melhor solução encontrada para a instância MS_40_65 com $\alpha = 0.5$	80
Figura 35 – Grafo da Solução inicial para a instância ME_40_99.	81
Figura 36 – Grafo da melhor solução encontrada para a instância ME_40_99 com $\alpha = 0.8$	81
Figura 37 – Grafo da melhor solução encontrada para a instância ME_40_99 com $\alpha = 0.7$	82
Figura 38 – Grafo da melhor solução encontrada para a instância ME_40_99 com $\alpha = 0.6$	82
Figura 39 – Grafo da melhor solução encontrada para a instância ME_40_99 com $\alpha = 0.5$	83
Figura 40 – Grafo da Solução inicial para a instância MD_40_164.	83
Figura 41 – Grafo da melhor solução encontrada para a instância MD_40_164 com $\alpha = 0.8$	84

Figura 42 – Grafo da melhor solução encontrada para a instância MD_40_164 com $\alpha = 0.7$	84
Figura 43 – Grafo da melhor solução encontrada para a instância MD_40_164 com $\alpha = 0.6$	85
Figura 44 – Grafo da melhor solução encontrada para a instância MD_40_164 com $\alpha = 0.5$	85

Lista de tabelas

Tabela 1 – Principais algoritmos para o PCMC.	25
Tabela 2 – Metaheurísticas baseadas em trajetória mais conhecidas.	29
Tabela 3 – Metaheurísticas baseadas em população mais conhecidas.	29
Tabela 4 – Descrição das instâncias da classe C1.	47
Tabela 5 – Descrição das instâncias da classe C2.	48
Tabela 6 – Valor da solução inicial para cada instância.	51
Tabela 7 – Comparação dos resultados obtidos pelo VNS com os do método exato.	53
Tabela 8 – Comparação dos resultados obtidos pelo VNS com os da literatura.	56

Lista de quadros

Quadro 1 – Trabalhos com aplicações a problemas do mundo real.	20
Quadro 2 – Problemas correlatos ao PCMC	24
Quadro 3 – Localização dos vértices	48
Quadro 4 – Resultados do CPLEX com o parâmetro de limite máximo de tempo de execução.	49
Quadro 5 – Resultados do CPLEX com o parâmetro de tolerância do GAP = 20%.	50
Quadro 6 – Resultados do CPLEX com o parâmetro de tolerância do GAP = 10%.	50
Quadro 7 – Resultados gerais do VNS para as instâncias da classe C1.	52
Quadro 8 – Resultados gerais do VNS para as instâncias da classe C2.	52
Quadro 9 – Resultados da instância PQ_14_20 da classe C1.	67
Quadro 10 – Resultados da instância PQ_18_41 da classe C1.	68
Quadro 11 – Resultados da instância SF_21_38 da classe C1.	69
Quadro 12 – Resultados da instância MS_40_65 da classe C2.	70
Quadro 13 – Resultados da instância ME_40_99 da classe C2.	71
Quadro 14 – Resultados da instância MD_40_164 da classe C2.	72

Lista de algoritmos

1	ALGORITMO DE FLOYD-WARSHALL	26
2	<i>Variable Neighborhood Search</i> - VNS	30
3	VND	31
4	PERTUBAÇÃO	43
5	DESCIDA 1	43
6	DESCIDA 2	44
7	VNS-VND	45

Lista de abreviaturas e siglas

AG	Algoritmo Genético
CNDP	<i>Continuos Network Design Problem</i>
CNT	Confederação Nacional do Transporte
DNDP	<i>Discrete Network Design Problem</i>
GO	Goiás
GRASP	<i>Greedy Randomized Adaptive Search Procedure</i>
ILS	<i>Iterated Local Search</i>
Max.	Maximizar
Min.	Minimizar
MNDP	<i>Mixed Network Design Problem</i>
NDP	<i>Network Design Problem</i>
NTU	Associação Nacional de Empresas de Transportes Urbanos
OC	Otimização Combinatória
PCMC	Problema do Caminho Mais Curto
PO	Pesquisa Operacional
PSO	<i>Particle Swarm Optimization</i>
SA	<i>Simulated Annealing</i>
SAA	<i>Sensitivity Analysis Algorithm</i>
TRDP	<i>Turning Restriction Design Problem</i>
TDP	<i>Turning Design Problem</i>
VND	<i>Variable Neighborhood Descent</i>
VNS	<i>Variable Neighbourhood Search</i>

Sumário

1	INTRODUÇÃO	17
1.1	Motivação da Pesquisa	19
1.2	Objetivos	20
1.2.1	Objetivo Geral	20
1.2.2	Objetivos Específicos	20
1.3	Organização da Dissertação	21
2	FUNDAMENTAÇÃO TEÓRICA	22
2.1	Otimização Combinatória	22
2.2	Problema do Caminho Mais Curto	23
2.2.1	Algoritmo de Floyd-Warshall	25
2.3	Metaheurísticas	27
2.3.1	<i>Variable Neighborhood Search</i> - VNS	29
3	PROBLEMA DE PROJETO DE REDE	32
3.1	Introdução	32
3.2	Definição do Problema	33
3.3	Nova Formulação Matemática	35
3.4	Trabalhos Relacionados	36
4	ABORDAGEM PROPOSTA	38
4.1	Visão Geral	38
4.2	Representação da Instância	40
4.3	Metaheurística VNS Desenvolvida	41
4.3.1	Representação de uma Solução	41
4.3.2	Perturbação	42
4.3.3	Busca Local	42
4.3.4	Detalhamento da Implementação	44
5	EXPERIMENTOS E RESULTADOS	46
5.1	Instâncias do Problema	46
5.1.1	Classe C1 de Instâncias	46
5.1.2	Classe C2 de Instâncias	47
5.2	Modelo Matemático	48
5.3	VNS	51
5.4	Análises dos Resultados	53

5.4.1	Comparação do VNS com o Método Exato	53
5.4.2	Comparação do VNS com Metaheurísticas da Literatura	55
6	CONCLUSÕES E TRABALHOS FUTUROS	59
6.1	Conclusões	59
6.2	Trabalhos Futuros	60
	REFERÊNCIAS	61
	APÊNDICES	66
	APÊNDICE A – RESULTADOS EXPERIMENTAIS DO VNS-VND	67
	APÊNDICE B – GRAFOS QUE REPRESENTAM AS INSTÂNCIAS	73

1 Introdução

O transporte veicular, seja ele privado ou público, é um dos principais meios de mobilidade urbana utilizada pela sociedade. Segundo Carvalho e Pereira (2011), existe deterioração das condições de mobilidade da população nos grandes centros urbanos, e um dos maiores motivos desta é o aumento dos veículos individuais, tendo como resultado a redução de viagens no transporte público, promovendo um crescimento do congestionamento do trânsito urbano e, por consequência, os acidentes de trânsito até mesmo com vítimas e a poluição veicular.

Em conformidade com uma pesquisa divulgada pela Confederação Nacional do Transporte e a Associação Nacional de Empresas de Transportes Urbanos CNT e NTU (2017), o brasileiro gasta em média trinta e sete minutos para chegar ao trabalho por motivo de engarrafamentos urbanos. Essa situação acarreta em problemas nos sistemas de trânsitos atuais em muitas cidades, que não conseguem suportar a demanda do crescimento do transporte veicular.

Em princípio, os problemas referentes ao trânsito são as principais preocupações dos governantes, dos profissionais da área de planejamento urbano e da população, seja de forma direta ou indireta, visto que todos usufruem (de algum jeito) a malha viária urbana e passam a ser atingidos por essas situações (LACORTT; KRIPKA; KRIPKA, 2013). Atualmente, existem diversos *softwares* disponíveis para a simulação de tráfego urbano em malhas viárias que auxiliam ao projetista na avaliação de projetos de modificações do sistema viário (BEN-AKIVA *et al.*, 1998; FELLENDORF, 1994; JAYAKRISHNAN; MAHMASSANI; HU, 1994). Entretanto, é necessário realizar uma modelagem das alterações antes de serem avaliadas quanto aos benefícios ganhos.

Levando em consideração o problema supracitado, modelos, sistemas e métodos vem sendo estudado na literatura e utilizados para resolver esse tipo de problema conhecido também como “problema de projeto de redes” (NDP – *Network Design Problem*), como mostram os estudos desenvolvidos por Leblanc (1975), Foulds (1981), Long *et al.* (2010), Duarte (2012).

No NDP, o objetivo é escolher um subconjunto dentro de uma grande quantidade de possíveis projetos de modificações da infraestrutura de uma rede viária que aumente a fluidez no tráfego de veículos (ou seja, diminua o nível de congestionamento) em uma rede viária (LEBLANC, 1975). Os projetos escolhidos para uma possível implantação na rede viária devem gerar um maior benefício no fluxo do trânsito, e o custo de implantação de um projeto não pode ultrapassar o orçamento pré-definido para a modificação da rede viária.

O problema de projeto de rede pode ser classificado de três formas: (i) discreta, que na literatura é conhecida como DNDP (*Discrete Network Design Problem*), e visa lidar com decisões discretas de projeto, como a construção de novas estradas, construção de novas faixas, a determinação das direções de ruas em sentido único e a determinação das restrições de viragem nas interseções; (ii) contínua, conhecida também como CNDP (*Continuos Network Design Problem*), que se preocupa com decisões contínuas, como expandir a capacidade das ruas, programar os semáforos e determinar pedágios para algumas ruas específicas; (iii) mista, que é MNDP (*Mixed Network Design Problem*), onde aborda ambos os tipos de decisões, a discreta e contínua (FARAHANI *et al.*, 2013).

Na forma contínua do NDP, uma das formas de otimização consideradas por pesquisadores de planejamento urbano bem como do comportamento do tráfego, é a sincronização de tempos do semáforo. Há diversos fatores na configuração destes dispositivos, tais como o tamanho do ciclo¹, os tempos das fases² e o tempo de compensação³, bem como a sincronização entre esses fatores, associados à influência que esta sincronização ocasiona sobre o fluxo total em uma malha viária (DUARTE, 2012).

Já a implantação de projetos que lidam em determinar a direção de algumas vias em sentido único (ou seja, tornar uma via em mão única) na malha viária, é uma das formas de modificação da infraestrutura da rede viária original, que resulta em melhorias no fluxo do trânsito de modo geral, dependendo de quais vias forem escolhidas para se tornar mão única. Por exemplo, quando se tem vias de mão dupla congestionadas, o tempo de travessia nestas vias são alto. Ao se realizar algumas modificações na malha viária, escolhendo uma configuração ótima (ou melhor), das direções de algumas vias em sentido único, existe a possibilidade de que as vias que estavam congestionadas por serem de mão dupla e não suportar o fluxo se tornem vias de mão única, melhorando o fluxo de modo geral na malha viária, conforme é mostrado por (DREZNER; WESOLOWSKY, 1997), (DREZNER; SALHI, 2000) e (DREZNER; SALHI, 2002).

O problema abordado no paragrafo anterior, é um caso particular do NDP, proposto por Drezner e Wesolowsky (1997), que visa encontrar a melhor configuração de vias de mão única e mão dupla na malha viária que minimize o tempo total de viagem de todos os usuários, e esse problema, se encaixa na classe dos problemas de projeto de rede com decisões discretas.

Considerando o que foi evidenciado, este trabalho visa conceber uma nova formulação matemática para o problema de projeto de rede com decisões discretas para determinar

¹ O “ciclo” de um semáforo é uma sequência finita de estados (geralmente, verde, amarelo e vermelho) que o mesmo percorre iterativamente.

² O “tempo de uma fase” do semáforo é uma quantidade de tempo que o mesmo permanece verde, amarelo ou vermelho durante um ciclo.

³ O “tempo de compensação” é o tempo transcorrido do momento em que um semáforo se torna verde até o próximo semáforo (por exemplo, em uma avenida) também entrar nesta mesma fase.

uma configuração ótima da malha viária com vias de mão única e mão dupla. Ainda faz parte deste trabalho, um estudo experimental com o método de solução proposta para o NDP. O método de solução proposta consiste na utilização do algoritmo de Floyd-Warshall para encontrar os caminhos mais curtos para todos os viajantes na malha viária entre origens e destinos, e na aplicação da metaheurística *Variable Neighborhood Search* - VNS (MLADENOVIĆ; HANSEN, 1997) ao NDP com a finalidade de apresentar soluções boas e aumentar o número de pesquisas da aplicação desta metaheurística a esse tipo de problema.

1.1 Motivação da Pesquisa

Com o avanço das tecnologias, muitos sistemas, modelos e métodos/técnicas tem sido desenvolvido para a resolução de problemas do mundo real. A maioria desses problemas, estão relacionados a áreas da Otimização Combinatória - OC, Pesquisa Operacional - PO, Ciência da Computação, Engenharias e outras, e cada vez mais, pesquisadores tem avançado ainda mais nas pesquisas, acarretando em resultados notáveis a problemas do mundo real, melhorando a qualidade de vida das pessoas de forma social, econômica e científica, tudo isso através do crescimento tecnológico e computacional ocorrido nos últimos anos. Diante disso, o Quadro 1, apresenta alguns trabalhos da literatura que trouxeram grandes contribuições ao resolverem problemas do mundo real relacionado com as áreas destacadas.

Os problemas relacionados a PO e OC, estão sempre se reformulando devido as suas derivações e variações. Isso é uma característica interessante, visto que é possível criar novas formulações matemáticas de acordo com os novos problemas, implementar métodos/técnicas aplicado ao problema, realizar experimentos e compará-los com sua abordagem exata se possível, pois a maioria destes problemas são pertencentes a classe de problemas NP-Difícil que tem uma alta complexidade computacional.

Sucintamente, os seguintes fatos relevantes motivam o desenvolvimento do presente trabalho:

- Propor uma nova formulação matemática para o problema de projeto de rede proposto inicialmente por Drezner e Wesolowsky (1997);
- Propor a aplicação de uma metaheurística para o problema projeto de rede;
- Aplicar em instâncias reais, com a finalidade de melhorar os problemas referente ao trânsito.

Quadro 1 – Trabalhos com aplicações a problemas do mundo real.

Descrição do Problema/Proposta	Referência
Apresenta um método heurístico baseado em sucessivas aproximações para resolver o problema do projeto de conversões (TDP – <i>Turning Design Problem</i>), o mesmo validou o método em uma rede viária bastante conhecida na literatura e também em uma região de Goiânia - GO, no Brasil.	Duarte (2012)
Proposta da aplicação dos problemas de localização de facilidades por p-medianas com o propósito de determinar o posicionamento mais eficiente de postos de entrega voluntária de materiais recicláveis na região central do município de Passo Fundo, aliado à aplicação do problema do caixeiro viajante que visa modelar o roteiro mais econômico para coleta e destinação destes resíduos.	Barão, Kripka e Kripka (2013)
Proposta de um método de solução para o problema de roteamento de veículos para a distribuição de alimentos congelados de múltiplos produtos, levando em consideração múltiplos fatores. Houve a validação do método desenvolvido aplicado em uma companhia de distribuidora de alimentos congelados na cidade de Pequim.	Zhang e Chen (2014)
Proposta de um modelo matemático para minimizar o número de motoristas necessários para cobrir um conjunto de viagens de exportação, importação e entre cidades, agrupando viagens que possuem menor deslocamento e realizar um planejamento de uma sequência de viagens para cada motorista.	Rodrigues e Santos (2018)
Proposta de uma metaheurística para o problema de alocação e sequenciamento de unidades de resgate, tendo como base dados referentes ao desastre súbito ocorrido na Região Serrana do Rio de Janeiro, no ano de 2011.	Cunha <i>et al.</i> (2018)

Fonte – Elaborado pelo autor.

1.2 Objetivos

1.2.1 Objetivo Geral

Este trabalho tem como objetivo geral desenvolver uma nova formulação matemática para o problema de projeto de rede com decisões discretas para determinar uma configuração ótima da malha viária com vias de mão única e mão dupla, e também, a implementação da metaheurística VNS para resolução do problema.

1.2.2 Objetivos Específicos

Para alcançar o objetivo geral, este trabalho tem os seguintes objetivos específicos:

1. Realizar o levantamento bibliográfico sobre o estado da arte do NDP e da metaheu-

- rística proposta;
2. Definir o problema proposto e propor a nova formulação matemática correspondente;
 3. Implementar a metaheurística proposta para o problema;
 4. Propor instâncias para validar o método proposto;
 5. Realizar experimentos computacionais;
 6. Analisar os resultados computacionais obtidos para verificar o desempenho da metaheurística proposta aplicado ao problema.

1.3 Organização da Dissertação

Além deste capítulo, o presente trabalho é composto por mais cinco capítulos. O Capítulo 2 apresenta a fundamentação teórica, que contém assuntos sobre otimização combinatória; problema do caminho mais curto; e metaheurísticas. No Capítulo 3 é discutido o problema de projeto de rede, bem como uma nova formulação matemática proposta neste trabalho. No Capítulo 4 é exposto o método de solução proposta desenvolvido, mostrando a visão geral da abordagem, e também, os detalhes de como foi a implementação do mesmo. Os experimento computacionais realizados e as análises dos resultados obtidos são discutidos no Capítulo 5. Por último, o Capítulo 6 aponta as conclusões e trabalhos futuros acerca deste trabalho.

2 Fundamentação Teórica

Este capítulo visa abordar uma fundamentação teórica sobre conceitos essenciais para o desenvolvimento deste trabalho. Na Seção 2.1 é apresentada uma breve introdução sobre otimização combinatória e seus problemas. A Seção 2.2 mostra o problema do caminho mais curto, bem como a sua formulação matemática, e uma explanação de um algoritmo para resolver este problema. Por fim, na Seção 2.3 são discutidos conceitos relacionados a metaheurística, assim como o método pesquisa em vizinhança variável (*Variable Neighborhood Search* - VNS).

2.1 Otimização Combinatória

De modo geral, a OC se concentra em minimizar ou maximizar uma função objetivo respeitando um domínio finito. Este domínio finito está associado a uma grande lista de possibilidades de soluções, e também a restrições que devem ser levadas em considerações durante a otimização em um problema (LEE, 2004).

Os problemas de otimização combinatória emergem em grande quantidade na vida real e científica, com aplicações em diversas áreas, tais como transportes, saúde, telecomunicações, finanças e entre outras (ARENALES *et al.*, 2007). Em Ausiello *et al.* (2003) são mostrados diversos exemplos destes problemas. Segue uma lista com os nomes de alguns problemas clássicos:

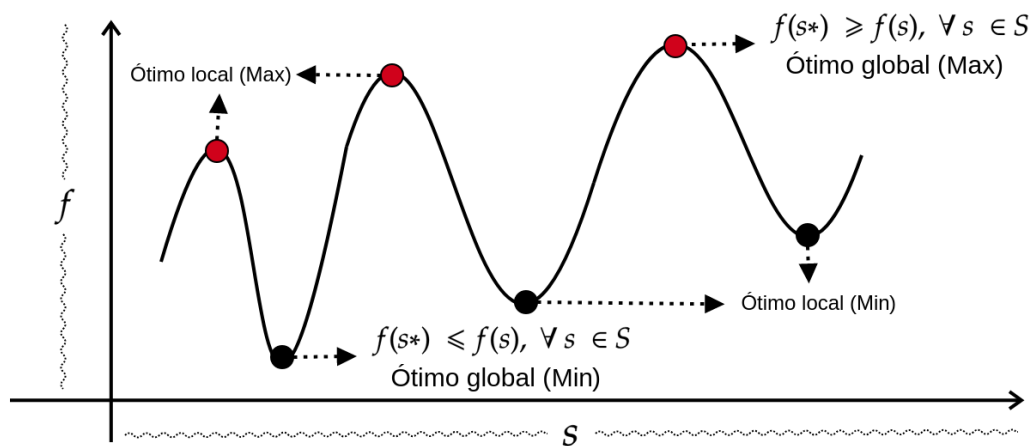
- Problemas da Mochila;
- Problemas de Cortes;
- Problemas de Designação;
- Problemas de cobertura, partição e empacotamento de conjuntos;
- Problemas de caixeiro-viajante;
- Problemas de carteiro chinês.

Segundo Resende e Silva (2013), um problema de otimização combinatória, pode ser definido por um conjunto finito $E = \{1, \dots, n\}$, um conjunto de soluções viáveis $S \subseteq 2^E$ e uma função objetivo $f : 2^E \rightarrow \mathbb{R}$, sendo todos definidos para cada problema específico. Estes problemas podem ter duas formas de objetivos diferentes, onde a primeira é a de maximização, na qual o objetivo consiste em encontrar uma solução ótima $s^* \in S$ tal que

$f(s^*) \geq f(s), \forall s \in S$, ou a segunda que é de minimização, na qual o objetivo é encontrar a solução ótima $s^* \in S$ tal que $f(s^*) \leq f(s), \forall s \in S$.

Para um melhor entendimento do que foi dito no parágrafo anterior, a Figura 1 ilustra graficamente em um plano cartesiano, a busca por encontrar soluções ótimas (ou seja, o ótimo global para um problema combinatório de minimização (*Min*) ou maximização (*Max*)). É possível que as vezes o método/técnica de solução para o problema encontre soluções que estão em ótimos locais, como mostra ainda na Figura 1. Uma solução de um ótimo local é considerada uma solução “boa” ou “próxima” em relação a solução ótima, e dependendo do problema ela é vista como relevante.

Figura 1 – Ilustração de soluções em problemas de otimização combinatória.



Fonte – Elaborada pelo autor.

Apesar do grande avanço de armazenamento e processamento computacional ocorrido nos últimos anos, a utilização de métodos exatos para encontrar soluções ótimas a problemas combinatórios com um grau de complexidade alta se torna inviável, devido a fatores como o tempo de execução, armazenamento e processamento. Sendo assim, muitos problemas que ocorrem na prática se resolvem através de métodos heurísticos que produzem rapidamente soluções de boa qualidade sem a garantia de alcançar o ótimo (RESENDE; SILVA, 2013). Neste contexto, muitos procedimentos de soluções a esse tipo de problemas se baseiam nas metaheurísticas que será explanado na Seção 2.3.

2.2 Problema do Caminho Mais Curto

O Problema do Caminho Mais Curto (PCMC) é um dos problemas mais conhecidos e antigos da pesquisa operacional. Considerando i e j dois vértices de um grafo $G = (V, A)$, o caminho mais curto entre i e j é uma sequência de arestas que conecta o vértice de origem i até o destino j sem repetir nenhum vértice durante este percurso, associando à

este um menor custo. É denominado um custo entre dois vértices i e j , representado por C_{ij} , o menor caminho existente entre eles. Caso não haja um caminho entre dois vértices quaisquer i e j , é considerado que o custo C_{ij} é infinito.

Segundo Goldbarg e Luna (2005) o problema do caminho mais curto está diretamente associado à soluções de diversos problemas de otimização combinatória, tais como os de roteamento, programação e sequenciamento de tarefas e outros. No Quadro 2, são apontados alguns trabalhos que abordaram problemas correlatos ao PCMC. Ainda neste contexto, o PCMC está também associado ao problema deste trabalho, que será explicado no Capítulo 3.

Quadro 2 – Problemas correlatos ao PCMC

Referências	Variantes
Loui (1983)	Problema do caminho mais curto em grafos estocásticos
Desrochers e Soumis (1988)	Problema do caminho mais curto com Janelas de Tempo
Yu e Yang (1998)	Problema do caminho mais curto robusto
Garfinkel, Fernández e Lowe (2006)	Problema do caminho mais curto k-Centro

Fonte – Elaborado pelo autor.

No trabalho desenvolvido por Orden (1956), ele observou que o problema do caminho mais curto é um caso especial de um problema de transbordo, e que o mesmo pode ser resolvido por programação linear. Então, a formulação matemática para o PCMC apresentada a seguir tem como base o modelo exposto por Orden (1956), e as notações utilizada pela formulação serão definidas antecipadamente para um melhor entendimento.

- V : conjunto de vértices;
- A : conjunto de arcos;
- s : representa um vértice de origem;
- t : representa um vértice de destino;
- i, j : índices dos vértices, $i, j = \{1, \dots, n\}$, onde $n = |V|$;
- c_{ij} : variável de custo para se deslocar do vértice i para o vértice j ;
- x_{ij} : variável booleana equivalente a $\begin{cases} 1, & \text{se o arco } (i, j) \text{ é selecionada,} \\ 0, & \text{caso contrário.} \end{cases}$

A formulação matemática para encontrar o caminho mais curto a partir do vértice s para o vértice t , é apresentada a seguir:

$$\text{Minimizar } Z = \sum_{(i,j) \in A} c_{ij} x_{ij} \quad (2.1)$$

$$\text{Sujeito à: } \sum_{(i,j) \in A} x_{ij} - \sum_{(j,i) \in A} x_{ji} = \begin{cases} 1, & \text{se } i = s \\ -1 & \text{se } i = t \\ 0, & \text{caso contrário} \end{cases}, \quad \forall i \in V \quad (2.2)$$

$$x_{ij} \in \{0, 1\}, \quad \forall (i, j) \in A \quad (2.3)$$

onde, a Equação 2.1 diz respeito a função objetivo do PCMC, que é minimizar o custo total de um caminho de uma origem ao um destino. A Equação 2.2, refere-se as restrições de conservação de fluxo, que garantem a conectividade a partir de um vértice de origem s para um vértice de destino t . Quanto a Equação 2.3, esta define o domínio da variável x_{ij} , como já mencionado logo acima.

Os métodos existentes para resolver o problema do caminho mais curto por meio de programação linear ficaram parcialmente de lado, visto que, há algoritmos alternativos mais simples e eficientes na literatura (ARENALES *et al.*, 2007). Na Tabela 1 são destacados os principais algoritmos para resolução do PCMC.

Tabela 1 – Principais algoritmos para o PCMC.

Referências	Algoritmos
Ford Jr. (1956), Moore (1959), Bellman (1958)	Algoritmo de Ford-Moore-Bellman
Dijkstra (1959)	Algoritmo de Dijkstra
Floyd (1962)	Algoritmo de Floyd-Warshall
Hart, Nilsson e Raphael (1968)	Algoritmo A^*

Fonte – Elaborado pelo autor.

Para o método proposto de solução no presente trabalho, será utilizado o algoritmo de Floyd-Warshall para encontrar os caminhos mais curto, devido ser de fácil entendimento e simples de implementação.

2.2.1 Algoritmo de Floyd-Warshall

O algoritmo de Floyd-Warshall foi publicado pela primeira vez na década de 60, por Floyd (1962) e Warshall (1962). Ingerman (1962) descreve em seu trabalho o procedimento deste algoritmo que é utilizado nos dias atuais. Este algoritmo é um exemplo de programação dinâmica que resolve problemas combinando as soluções para subproblemas. De acordo com Cormen *et al.* (2012), um algoritmo de programação dinâmica resolve cada subproblema uma única vez e depois guarda sua resposta em uma tabela, evitando assim a necessidade de recalculá-la toda vez que resolver cada sub-subproblema.

O algoritmo de Floyd-Warshall encontra os caminhos mais curtos entre todos os pares de vértices em um grafo e o mesmo funciona com arcos com custos negativos, mas não permite circuitos¹ com custos negativos.

Considere um grafo $G = \{V, A\}$ com n vértices V . Denota-se $D^k(i, j)$ como o custo do caminho mais curto entre i e j podendo-se utilizar apenas os vértices de 1 a k como vértices intermediários². $D^0(i, j)$ é o caminho mais curto de i a j sem usar nenhum vértice intermediário, ou seja, $D^0(i, j) = c_{ij}$, caso o arco $(i, j) \in A$, caso contrário, $D^0(i, j) = +\infty$, e $D^0(i, i) = 0$ para todo $i = 1, \dots, n$. Veja que, $D^k(i, j) = +\infty$ se não há caminho entre os vértices i e j tendo como vértices intermediários de 1 a k . A partir disso, tem-se que:

- Se $D^{k+1}(i, j)$ não usa o nó $k + 1$ como intermediário, então:

$$D^{k+1}(i, j) = D^k(i, j) \quad (2.4)$$

- Se $D^{k+1}(i, j)$ usa o nó $k + 1$ como intermediário, então:

$$D^{k+1}(i, j) = D^k(i, k + 1) + D^k(k + 1, j) \quad (2.5)$$

portanto, o valor de $D^{k+1}(i, j)$ é dada pela fórmula abaixo.

$$D^{k+1}(i, j) = \min\{D^k(i, j), D^k(i, k + 1) + D^k(k + 1, j)\} \quad (2.6)$$

Algoritmo 1: ALGORITMO DE FLOYD-WARSHALL

```

1  inicio
2  |   para  $i = 1, \dots, n$  faça
3  |   |   para  $j = 1, \dots, n$  faça
4  |   |   |    $D^0(i, j) \leftarrow c_{ij}$ 
5  |   |   fim
6  |   fim
7  |   para  $k = 1, \dots, n$  faça
8  |   |   para  $i = 1, \dots, n$  faça
9  |   |   |   para  $j = 1, \dots, n$  faça
10 |   |   |   |    $D^k(i, j) \leftarrow \min\{D^{k-1}(i, j), D^{k-1}(i, k) + D^{k-1}(k, j)\}$ 
11 |   |   |   fim
12 |   |   fim
13 |   fim
14 fim

```

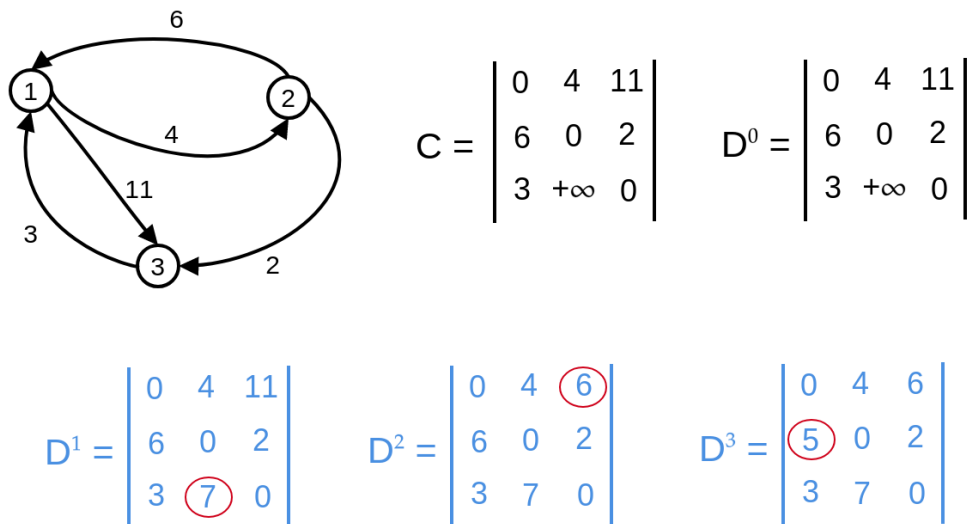
A Fórmula 2.6 é utilizada pelo algoritmo de Floyd-Warshall, e mostra que é possível determinar o $D^{k+1}(i, j)$ à partir do uso do resultado de uma iteração anterior que é

¹ Um circuito em um grafo é caminho simples que começa em um vértice i e termina nele mesmo. Um caminho simples não há repetição de arcos entre dois vértices quaisquer.

² Um vértice intermediário em um caminho de i a j , é qualquer vértice que faz parte deste caminho, exceto os vértices i e j (CORMEN *et al.*, 2012).

$D^k(i, j)$, $D^k(i, k + 1)$ e $D^k(k + 1, j)$. Com essa fórmula resolve-se o problema do caminho mais curto entre todos os pares de vértices, que é descendente da aplicação da técnica de programação dinâmica (ARENALLES *et al.*, 2007). O Algoritmo 1 apresenta o pseudocódigo para a resolução do PCMC para todos os pares de vértices em um grafo com base nos procedimentos mencionados.

Figura 2 – Ilustração do procedimento de execução do algoritmo de Floyd-Warshall.



Fonte – Elaborada pelo Autor.

Na Figura 2, é mostrado como seria a execução deste algoritmo no grafo ilustrado na mesma figura, nota-se que há atualização na matriz D devido a utilização dos vértices intermediários a cada iteração para encontrar os caminhos mais curtos.

2.3 Metaheurísticas

A utilização de métodos exatos para resolução de problemas de OC que são NP-difíceis tem uma alta complexidade e exigem tempo super polinomial (por exemplo, tempo exponencial) para serem resolvidos (CORMEN *et al.*, 2012). Com isso, uma das maneiras de resolução destes problemas em um tempo viável é a utilização de metaheurísticas.

Uma metaheurística pode ser considerada como uma estrutura de algoritmo genérico que pode ser aplicada a diferentes problemas de otimização com a realização de algumas modificações para o problema específico (GLOVER; KOCHENBERGER, 2003). Para Brownlee (2011) as metaheurísticas são uma extensão de heurísticas, combinando um ou mais métodos heurísticos, com uma alta estratégia de buscar soluções. Nestes métodos, existem componentes estocásticos que permite fugir de ótimos locais durante o procedimento de buscar soluções em problemas de otimização (CHOPARD; TOMASSINI, 2018).

No trabalho desenvolvido por Blum e Roli (2003), eles apresentam nove propriedades que caracterizam as metaheurísticas, que são as seguintes:

- Metaheurísticas são estratégias que guiam um processo de busca;
- O objetivo é explorar eficientemente o espaço de busca, a fim de encontrar soluções próximas do ótimo;
- As técnicas na qual constituem uma metaheurística variam de procedimentos simples de busca local a processos complexos de aprendizagem;
- Metaheurísticas são métodos aproximativos e usualmente não determinísticos;
- Elas podem incorporar mecanismos para evitar ficar preso em áreas confinadas (em ótimos locais) do espaço de busca;
- Os conceitos básicos das metaheurísticas permitem uma descrição abstrata de nível;
- Metaheurísticas não são específicas dos problemas;
- Metaheurísticas podem fazer uso de conhecimento específico de domínio na forma de heurísticas que são controladas por uma estratégia de alto nível;
- As metaheurísticas mais avançadas de hoje usam a experiência de busca (incorporada em alguma forma de memória) para guiar a busca.

Para Blum e Roli (2003), há dois fatores em uma metaheurística que são de grande importância que é a diversificação e a intensificação. O termo diversificação, refere-se a exploração do espaço de busca, já a intensificação refere-se a exploração na experiência da busca acumulada. Em outras palavras, a diversificação visa investigar soluções em diversas regiões diferentes do espaço de busca em um problema, e a intensificação utiliza informações de buscas anteriores para aprofundar em regiões promissoras do espaço de busca.

Na literatura especializada, existem algumas classificações de metaheurísticas, normalmente divididas em dois grupos: (i) métodos baseados em trajetória, e (ii) métodos baseados em população. Alguns trabalhos também tem utilizado metaheurísticas baseadas em método híbrido, que são técnicas que visam agrupar diferentes técnicas de otimização (LOZANO; BLUM, 2010; GASPERO *et al.*, 2010).

O grupo de metaheurísticas baseadas em trajetória busca encontrar uma solução ótima através da manipulação de uma única solução ao longo da execução do algoritmo. Eles partem de uma solução inicial, e evolui ao longo do algoritmo seguindo um certo mecanismo (ou seja, uma trajetória) até que o critério de parada seja alcançado (HÉLIODORE *et al.*, 2017). A Tabela 2 lista as metaheurísticas mais conhecidas baseadas em trajetória.

Tabela 2 – Metaheurísticas baseadas em trajetória mais conhecidas.

Metaheurísticas	Referências
<i>Simulated Annealing</i> - SA	Kirkpatrick, Gelatt e Vecchi (1983)
Busca Tabu	Glover (1986)
Greedy Randomized Adaptive Search Procedure - GRASP	Feo e Resende (1995)
Busca Local Guiada	Voudouris e Tsang (1996)
<i>Variable Neighborhood Search</i> - VNS	Mladenović e Hansen (1997)
<i>Iterated Local Search</i> - ILS	Stützle (1998) e Lourenço, Martin e Stützle (2001)

Fonte – Elaborado pelo autor.

As metaheurísticas baseadas em população são caracterizadas pela manipulação de um conjunto de soluções pré-definida chamada de população. Para alcançar uma solução de boa qualidade, estes algoritmos manipulam várias soluções que exploram o espaço de busca de acordo com suas próprias informações, bem como as informações que outras soluções transmitem (HÉLIODORE *et al.*, 2017). As metaheurísticas baseadas em população mais conhecidas são apresentadas na Tabela 3.

Tabela 3 – Metaheurísticas baseadas em população mais conhecidas.

Metaheurísticas	Referências
Algoritmo Genético - AG	Holland (1975) e Goldberg (1989)
Algoritmo Memético	Moscato (1989)
Colônia de formigas	Dorigo (1992)
<i>Particle Swarm Optimization</i> – PSO	Kennedy e Eberhart (1995)
Evolução Diferencial	Storn e Price (1995)

Fonte – Elaborado pelo autor.

No presente trabalho é exposto uma metaheurística baseada em trajetória na próxima seção, visto que a mesma faz parte do artefato do estudo proposto.

2.3.1 *Variable Neighborhood Search* - VNS

A *Variable Neighborhood Search* (VNS) também conhecida como pesquisa (ou busca) em vizinhança variável. É uma metaheurística proposta por (MLADENOVIC; HANSEN, 1997), que se baseia na aplicação de uma simples estratégia, que é a mudança nas estruturas de vizinhança de forma sistemática (BLUM; ROLI, 2003).

Para Brownlee (2011), a estratégia do VNS é motivada por três princípios:

1. Um mínimo local para uma estrutura de vizinhança pode não ser um mínimo local para uma estrutura de vizinhança diferente;

2. Um mínimo global é um mínimo local para todas as estruturas de vizinhança;
3. Mínimos locais são relativamente próximos de mínimos globais para muitos problemas.

A VNS foi aplicada com sucesso para resolver diversos problemas, como o da p-mediana (GARCÍA-LÓPEZ *et al.*, 2002), o problema de roteamento com várias estações (POLACEK *et al.*, 2004) e muitos outros problemas clássicos (MLADENOVIC; HANSEN, 1999; HANSEN; MLADENOVIC, 2002; HANSEN; MLADENOVIC, 2003).

A busca em vizinhança variável possui em seu principal ciclo três etapas: (i) *Shaking* ou perturbação que visa mudar a solução corrente com algum tipo de mecanismo, geralmente, essa etapa escolhe uma solução vizinha da solução corrente de forma aleatória; o (ii) *Descend* é um procedimento de busca local aplicado na solução após a realização da perturbação; e (iii) *move* está ligado a condição de mudança da vizinhança após a busca local (BLUM; ROLI, 2003). O Algoritmo 2 apresenta o pseudocódigo do VNS tradicional com as etapas mencionadas.

Algoritmo 2: *Variable Neighborhood Search - VNS*

```

1 inicio
2    $r \leftarrow \text{numeroVizinhanca}()$ 
3    $s \leftarrow \text{solucaoInicial}()$ 
4   enquanto Critério de parada não satisfeito faça
5      $k \leftarrow 1$  // Tipo de estrutura de vizinhança
6     enquanto  $k \leq r$  faça
7       //  $N(k)$  refere-se a estrutura de vizinhança  $k$ 
8        $s' \leftarrow \text{escolherVizinho}(N(k), s)$  // Perturbação
9        $s'' \leftarrow \text{buscaLocal}(s')$ 
10      se  $f(s'') < f(s)$  então
11         $s \leftarrow s''$ 
12         $k \leftarrow 1$ 
13      fim
14      senão
15         $k \leftarrow k + 1$ 
16      fim
17    fim
18 fim
    retorna:  $s$  // Melhor solução encontrada

```

A VNS apresentada no Algoritmo 2, de forma resumida, parte de uma solução inicial qualquer, a cada iteração seleciona-se aleatoriamente um vizinho dentro da vizinhança da solução corrente, e passa por um procedimento de busca local. Se a solução ótima local, s'' , for melhor que a solução corrente s , a busca continua de s'' começando da primeira estrutura de vizinhança. Caso contrário, a busca continua a partir do próximo vizinho. Quando a condição de parada for atingida o processo é encerrado.

Algoritmo 3: VND

```
Entrada:  $s$ 
1 inicio
2    $r \leftarrow \text{numeroVizinhanca}()$ 
3    $s \leftarrow \text{solucaoInicial}()$ 
4   enquanto  $k \leq r$  faça
5     //  $N(k)$  refere-se a estrutura de vizinhança  $k$ 
6      $s' \leftarrow \text{encontrarMelhorVizinho}(N(k), s)$ 
7     se  $f(s') < f(s)$  então
8        $s \leftarrow s'$ 
9        $k \leftarrow 1$ 
10    fim
11  senão
12     $k \leftarrow k + 1$ 
13  fim
14 fim
    retorna:  $s$  // Melhor solução encontrada
```

No trabalho de Mladenović e Hansen (1999) é proposto um método de descida em vizinhança variável (*Variable Neighborhood Descent* - VND) que é um método de busca local e pode ser utilizado no VNS. O VND tem a ideia de utilizar uma estrutura de vizinhança enquanto for possível melhorar a solução atual, caso contrário, troca a vizinhança e realiza novamente o procedimento. O Algoritmo 3 ilustra o pseudocódigo deste método.

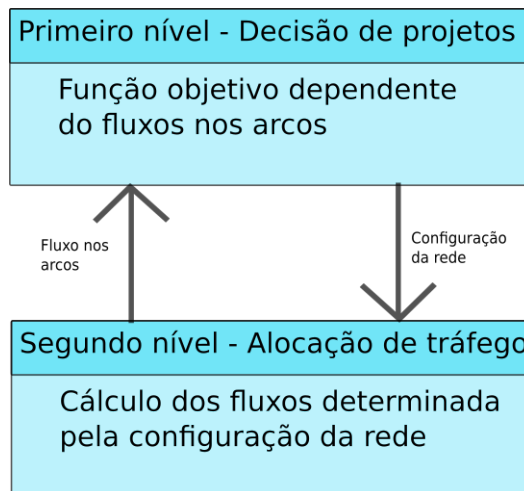
3 Problema de Projeto de Rede

Este capítulo discute sobre o objeto de estudo proposto neste trabalho. Na Seção 3.1 é mostrada uma breve introdução sobre o problema de projeto de rede. A Seção 3.2, define formalmente o problema de estudo do presente trabalho. Uma nova formulação matemática para o problema de projeto de rede é apresentada na Seção 3.3. Por fim, alguns trabalhos relacionados são descritos resumidamente na Seção 3.4.

3.1 Introdução

O problema de projeto de rede (*Network Design Problem - NDP*) foi proposto pela primeira vez por (LEBLANC, 1975). De acordo com o mesmo, este problema consiste designar um subconjunto dentro de uma grande quantidade de possíveis projetos de modificações da infraestrutura de uma rede viária que aumente a fluidez no tráfego de veículos (ou seja, diminua o nível de congestionamento) em uma rede viária. O NDP é um problema de otimização combinatória classificado como NP-difícil (CHUZHOY *et al.*, 2008).

Figura 3 – Ilustração do NDP como um problema de dois níveis.



Fonte – Adaptada de Duarte (2012)

Nas últimas décadas, tem surgido diversos modelos variantes e métodos propostos para o NDP. Na literatura, muitos trabalhos tem proposto modelar o problema de projeto de rede como um problema de programação não linear em dois níveis e é NP-Difícil (GAO; WU; SUN, 2005), a ilustração desta modelagem é mostrada na Figura 3. O primeiro nível (ou nível superior) tem o objetivo de minimizar a função objetivo que é o tempo total de viagem em uma malha viária, que depende da resolução do problema do segundo nível, que

é o problema de alocação de tráfego. A forma mais comum de se resolver este problema é partindo do princípio que o fluxo na rede fique em EU (Equilíbrio de Usuário) (ZHANG; GAO, 2007). Uma rede está em equilíbrio de usuário quando nenhum viajante consegue melhorar seu tempo de viagem entre uma origem e um destino, que parte de uma premissa de que os usuários escolhem as melhores rotas, que é um dos princípios de Wardrop (1952).

A proposta do problema de projeto de rede deste trabalho é modelado como um problema de programação linear que será mostrado na próxima seção. O mesmo é da família do problema NDP com decisões discretas (ou seja, DNDP - *Discrete Network Design Problem*) como já mencionado neste trabalho, e tem o objetivo de minimizar o tempo total da viagem em uma malha determinando uma configuração ótima (ou, melhor) da malha viária com vias de mão única e mão dupla.

3.2 Definição do Problema

A definição do problema de projeto de rede a ser abordada neste trabalho já supracitado, foi proposto por (DREZNER; WESOLOWSKY, 1997). O mesmo, propõe uma formulação matemática para resolução do problema por programação inteira, na qual, tem o objetivo escolher uma configuração ótima das vias de mão única e mão dupla na malha viária para minimizar o tempo total de viagem na mesma.

A formulação matemática a ser apresentada é adaptada de Drezner e Wesolowsky (1997), tendo a sua notação apropriada para ficar uniforme com a formulação proposta neste trabalho a ser apresentada nas próximas seções. As notações utilizadas pela formulação do problema serão detalhadas previamente para um melhor entendimento:

- n : número de vértices;
- m : número de arcos;
- V : conjunto de todos os vértices;
- A : conjunto de todos os arcos. Cada elemento de A é definido como um par (r, t) de nós. Cada arco pode ser uma via de mão dupla, a cardinalidade de A é $2m$;
- c_{rt} : definido para todo $(r, t) \in A$, é um custo de travessia no arco (r, t) , quando ele é mão dupla;
- α : fator pelo qual o custo de travessia é multiplicado quando o arco se torna uma via de mão única ($0 < \alpha < 1$). É possível ter diferentes valores de α para cada arco, mas para simplificar, assumimos o mesmo α para todos os arcos. Quando o tráfego é leve, α se aproxima de 1, e para o tráfego congestionado, α é pequeno;

- M : é um valor muito alto, o qual é multiplicado pelo custo de travessia do arco caso ele não esteja disponível;
- p_{rt} : para todo $(r, t) \in A$, é uma variável de implantação de projetos que define uma configuração da malha viária em vias de mão única. Se o arco de r para t não está disponível, $p_{rt} = 0$, e $p_{rt} = 1$ se o arco está disponível para travessia. Note que uma via não pode ser bloqueada em ambos os lados, assim, $p_{rt} + p_{tr} \geq 1$. Por exemplo, se $p_{rt} = 1 \forall (r, t) \in A$, significa que a rede consiste em vias de mão dupla;
- P : conjunto das variáveis p_{rt} de tamanho $2m$;
- w_{ij} : número de veículos que pretendem ir de i para j ;
- OD : conjunto de todos os veículos que desejam partir de uma origem para um destino;
- x_{ijrt} : é uma variável para encontrar o caminho mais curto de i para j , e está definida para $(r, t) \in A$ e $(i, j) \in OD$.
- $s_{ij}(P)$: é o custo do caminho mais curto de i para j para a rede definida por P ;
- $F(P)$: É a função objetivo para dado um P .

$$F(P) = \sum_{(i,j) \in OD} w_{ij} s_{ij}(P) \quad (3.1)$$

Para se encontrar $s_{ij}(P)$ deve-se resolver o problema do caminho mais curto, onde a formulação adotada pelos autores é possível ver no trabalho deles (DREZNER; WESOLOWSKY, 1997). Dado as notações apresentadas, a Equação 3.1 que representa a função objetivo do problema, e a adaptação de um modelo matemático do PCMC para $s_{ij}(P)$, a formulação matemática do NDP como um problema de programação inteira, é a

seguinte:

$$\text{Minimizar} \quad F(P) = \sum_{(i,j) \in OD} w_{ij} \left[\sum_{(r,t) \in A} x_{ijrt} (c_{rt} [\alpha + (1 - \alpha)p_{tr} + M(1 - p_{rt})]) \right] \quad (3.2)$$

$$\text{Sujeito à:} \quad \sum_{r \neq t} x_{ijrt} = \sum_{k \neq r} x_{ijtk} \quad \forall t, (r, t) \in A, (t, k) \in A, (i, j) \in OD \quad (3.3)$$

$$\sum_{(i,t) \in A} x_{ijit} = 1 \quad \forall (i, j) \in OD \quad (3.4)$$

$$\sum_{(t,j) \in A} x_{ijtj} = 1 \quad \forall (i, j) \in OD \quad (3.5)$$

$$p_{rt} + p_{tr} \geq 1 \quad \forall (r, t) \in A \quad (3.6)$$

$$p_{rt} \in \{0, 1\} \quad \forall (r, t) \in A \quad (3.7)$$

$$x_{ijrt} \geq 0 \quad (3.8)$$

Onde, a Equação 3.2 refere-se a função objetivo do NDP, que é minimizar o tempo total de viagem na malha viária. As Equações 3.3, 3.4 e 3.5 são responsáveis pela conservação de fluxo, que asseguram a conectividade para todas as origens e destinos pertencentes a OD. A Equação 3.6 garante que uma via da malha viária não seja bloqueada em ambos os lados com a implantação dos projetos. Por fim, o domínio das variáveis p_{rt} e x_{ijrt} são definidas pelas Equações 3.7 e 3.8, respectivamente.

3.3 Nova Formulação Matemática

Com a definição do problema e a modelagem matemática apresentada anteriormente, o presente trabalho propõe reformular a formulação matemática proposta por (DREZNER; WESOLOWSKY, 1997) de forma a torná-lo mais simples e compacta.

Inicialmente, foi percebido que este modelo poderia ser reduzido, visto que, na formulação utilizam-se três equações de restrições que servem para conservação de fluxo que são adaptadas do problema de caminho mais curto abordada por eles. Com isso, houve a ideia de se utilizar o modelo matemático do PCMC apresentado na Seção 2.2 para o $s_{ij}(P)$ da Equação 3.1, e realizar as adaptações necessárias para a criação/reformulação

da formulação matemática para o NDP proposta no presente trabalho, que é a seguinte:

$$\text{Minimizar} \quad F(P) = \sum_{(i,j) \in OD} w_{ij} \left[\sum_{(r,t) \in A} x_{ijrt} (c_{rt} [\alpha + (1 - \alpha)p_{tr} + M(1 - p_{rt})]) \right] \quad (3.9)$$

$$\text{Sujeito à:} \quad \sum_{(r,t) \in A} x_{ijrt} - \sum_{(t,r) \in A} x_{ijtr} = \begin{cases} 1, & \text{se } r = i \\ -1, & \text{se } r = j \\ 0, & \text{caso contrário} \end{cases}, \forall (i, j) \in OD, r \in A \quad (3.10)$$

$$p_{rt} + p_{tr} \geq 1 \quad \forall (r, t) \in A \quad (3.11)$$

$$p_{rt} \in \{0, 1\} \quad \forall (r, t) \in A \quad (3.12)$$

$$x_{ijrt} \in \{0, 1\} \quad \forall (i, j) \in OD, (r, t) \in A \quad (3.13)$$

onde as Equações 3.9, 3.11, 3.12 são as mesmas da formulação anterior. A Equação 3.10 é responsável pela conservação de fluxo que assegura a conectividade para todas as origens e destinos pertencentes a OD. Podemos notar que esta equação substitui as Equações 3.3, 3.4 e 3.5 do modelo anterior. Por último, o domínio da variável x_{ijrt} é definida pela Equação 3.13.

É importante destacar que este modelo não leva em consideração um limite de orçamento pré-estabelecido para a quantidade de implantação de projetos para a modificação da malha viária original, como alguns modelos encontrado na literatura para os problemas da classe NDP.

3.4 Trabalhos Relacionados

Drezner e Wesolowsky (1997), apresentam o problema de projeto de rede com decisões de determinar as direções das vias de mão única e mão dupla em uma malha viária, a fim de encontrar uma melhor configuração da malha viária de modo que minimize o tempo total de viagem na mesma. Para resolução do problema, eles utilizaram um método de *Branch and Bound*, heurísticas e a metaheurística *Simulated Annealing*. Os trabalhos de Drezner e Salhi (2000) e Drezner e Salhi (2002) abordaram o mesmo problema, mas o primeiro aplicou a Busca Tabu como método de solução, e o segundo usou as metaheurísticas Algoritmo Genético e *Simulated Annealing* como métodos de solução para o problema.

O trabalho desenvolvido por Leblanc (1975) mostra que é possível minimizar o congestionamento de uma malha viária aumentando a capacidade de algumas ruas. Para isso, foi utilizado *Branch and Bound*. Já no trabalho de Foulds (1981), para resolver o NDP, foi utilizado, também, o método exato *Branch and Bound*, e usa a proibição de algumas ruas para minimizar o fluxo da rede viária. Long *et al.* (2010), abordam um problema variante do NDP que é o problema de projeto de restrição de conversão (TRDP - *Turning*

Restriction Design Problem), que consiste em determinar um conjunto de restrições de conversões nas interseções das ruas que devem ser implementadas, de modo que, melhore o fluxo na rede viária. Os mesmos apresentaram um método de *Branch and Bound*, baseado em um algoritmo de análise de sensibilidade (SAA - *Sensitivity Analysis Algorithm*) para encontrar uma estratégia de restrição de conversão ótima.

O trabalho de Duarte (2012) apresenta um método heurístico baseado em sucessivas aproximações para resolver o problema do projeto de conversões (TDP – *Turning Design Problem*), que consiste em encontrar um conjunto de conversões permitidas e proibidas a partir de um conjunto de possíveis projetos de conversões que promovam uma melhoria no fluxo total de uma malha viária.

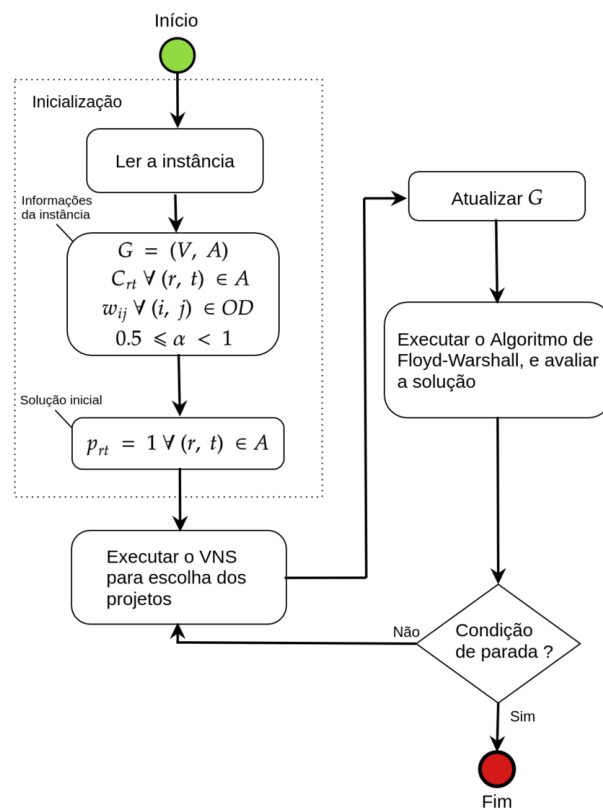
4 Abordagem Proposta

Este capítulo disserta sobre a abordagem proposta desenvolvida neste trabalho. Na Seção 4.1 é descrito de modo geral o método de solução proposto para o problema de projeto de rede. Na Seção 4.2 mostra como foi a forma utilizada para representar uma instância do problema deste trabalho. Por fim, na Seção 4.3 é apresentado com detalhes o desenvolvimento da metaheurística VNS, descrevendo e mostrando as partes necessárias deste método.

4.1 Visão Geral

Este trabalho propõe solucionar o Problema de Projeto de Rede por meio de uma abordagem iterativa, representada pelo fluxograma ilustrado na Figura 4, onde é necessário realizar algumas etapas com o objetivo de viabilizar uma solução para o problema como mostra a figura.

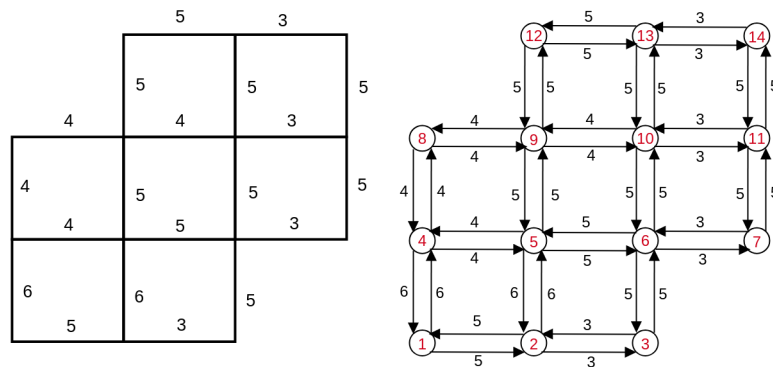
Figura 4 – Fluxograma do método de solução proposta.



Fonte – Elaborada pelo autor.

Inicialmente, o problema deve ser instanciado como um grafo, no qual os vértices representam as interseções entre as vias da malha viária e são rotulados de forma numérica, e os arcos correspondem as vias. Além disso, cada arco tem um peso que caracteriza um custo de travessia da via. A Figura 5 ilustra graficamente o que seria uma instância do problema, mostrando a malha viária de um lado e o grafo que a representa do outro lado.

Figura 5 – Ilustração da malha viária e o grafo que a representa.



Fonte – Adaptada de Drezner e Wesolowsky (1997).

Com o requisito fundamental citado anteriormente, inicializam-se as etapas de resolução do problema, começando pela etapa de leitura da instância (“Ler a instância”), que necessita de alguns dados sobre a mesma: um grafo $G = (V, A)$, onde V é o conjunto de vértices e A o conjunto dos arcos, que representa uma malha viária; uma matriz de origem e destino OD , que diz a quantidade de veículos a se deslocar de uma origem para um destino; uma matriz de custo do grafo; definir um valor para α ($0.5 \leq \alpha \leq 0.9$); e uma matriz de implantação de projetos (solução inicial do problema) com todas as vias sendo mão dupla, ou seja, $p_{rt} = 1 \forall (r, t) \in A$. Dadas estas informações, a fase de inicialização do método de solução proposto destacado no fluxograma da Figura 4 tem sido finalizada.

Após a fase de inicialização, dá-se início ao ciclo iterativo, que começará pela execução da metaheurística VNS desenvolvida para encontrar a escolha da implantação dos projetos P que é a solução do problema. Posteriormente, temos a fase de atualizar o grafo de acordo com os projetos escolhidos pela metaheurística anteriormente. O algoritmo de Floyd-Warshall apresentado na Seção 2.2.1, é utilizado para encontrar os caminhos mais curtos para todos os viajantes da matriz OD , e também é avaliada a solução atual do problema. Após isso, caso a condição de parada tenha sido satisfeita o processo acaba, caso contrário, ele volta para a etapa de execução do VNS. Quando acabar este processo de execução iterativo, temos uma malha viária com uma melhor configuração de vias de mão única e mão dupla de acordo com a implantação dos projetos encontrada pelo método. Nas próximas seções, será apresentado com detalhes o desenvolvimento do método proposto.

4.2 Representação da Instância

Foram utilizados arquivos de textos para representar a instância do problema. Conforme visto, uma instância tem informações sobre a malha viária e uma demanda de motoristas (ou viajantes) que desejam ir de uma origem a um destino na mesma. Por exemplo, para representar a instância ilustrada na Figura 5, são utilizados dois arquivos de texto que tem a seguinte estrutura:

- (i) O primeiro arquivo de texto possui informações sobre o grafo, e é empregado um formato simples para descrever o mesmo, onde a primeira linha do arquivo diz uma quantidade n de vértices que o grafo tem, e a segunda uma quantidade m de arestas. O arquivo contém mais m linhas que diz respeito a cada aresta do grafo, no seguinte formado “ $i \ j \ k$ ”, ou seja, o grafo possui arcos que começam do vértice i para o vértice j e tem um custo associado neste arco no valor igual a k . Logo abaixo, é possível ver o exemplo do arquivo que descreve o grafo da Figura 5.

1	<NODES> 14	22	7 11 5
2	<LINKS> 40	23	8 4 4
3	1 2 5	24	8 9 4
4	1 4 6	25	9 5 5
5	2 1 5	26	9 8 4
6	2 3 3	27	9 10 4
7	2 5 6	28	9 12 5
8	3 2 3	29	10 6 5
9	3 6 5	30	10 9 4
10	4 1 6	31	10 11 3
11	4 5 4	32	10 13 5
12	4 8 4	33	11 7 5
13	5 2 6	34	11 10 3
14	5 4 4	35	11 14 5
15	5 6 5	36	12 9 5
16	5 9 5	37	12 13 5
17	6 3 5	38	13 10 5
18	6 5 5	39	13 12 5
19	6 7 3	40	13 14 3
20	6 10 5	41	14 11 5
21	7 6 3	42	14 13 3

- (ii) O segundo arquivo dispõe informação sobre uma quantidade de motoristas que desejam se deslocar de uma origem a um destino, como já supracitado. O formato desta informação no arquivo é por meio de uma matriz quadrada de ordem n , onde n é a quantidade de vértices do grafo da instância. Cada elemento w_{ij} da matriz OD corresponde a uma demanda de veículos que desejam partir de uma origem i para um destino j . Suponha que para a instância que estamos exemplificando, há motoristas que desejam se deslocar de todas as origens para todos os destinos igual a 1, ou seja, $w_{ij} = 1$ para todo par de vértices onde $i \neq j$. Logo abaixo, é possível ver o exemplo do arquivo que descreve o que foi dito.

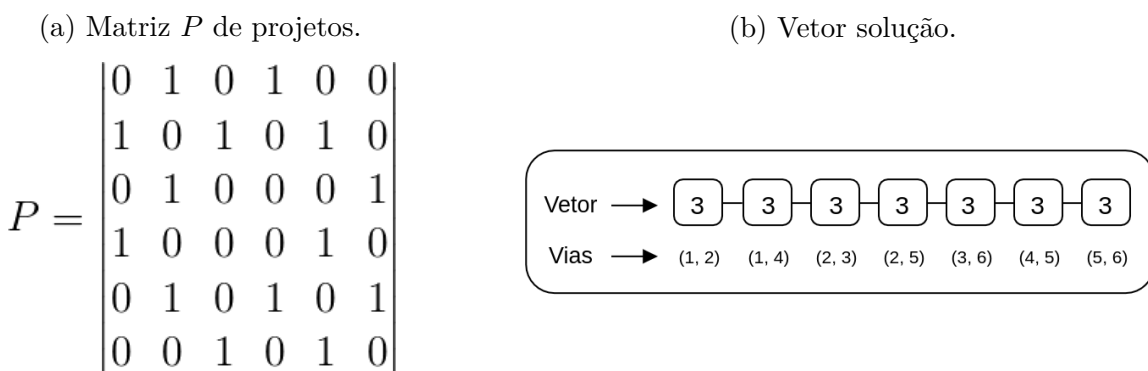
1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1
3	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1
4	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1
5	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1
6	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1
7	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1
8	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1
9	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1
10	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1
11	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1
12	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1
13	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1
14	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0

4.3 Metaheurística VNS Desenvolvida

4.3.1 Representação de uma Solução

Uma solução para o problema de projeto de rede é representada por uma matriz P de implantação de projetos na malha viária que indicará a configuração da mesma. Em conformidade com o que foi visto, se uma posição p_{ij} da matriz P é igual a 1 (um), quer dizer que a via (i, j) está disponível, se for igual a 0 (zero), a via (i, j) não está disponível. Note que a matriz P tem a estrutura de uma matriz de adjacência de um grafo, ou seja, com ela é possível moldar a configuração da malha viária com vias de mão única e mão dupla que é representada por um grafo. Por exemplo, a Figura 6a ilustra uma matriz P de projetos de uma parte da malha viária da Figura 5, considerando apenas os vértices de 1 a 6 e as arestas que estão ligando esses vértices entre si.

Figura 6 – Representações de uma solução.



Fonte – Elaborada pelo autor.

Computacionalmente, foi utilizado um vetor de inteiros para representar uma solução (matriz P de projetos) do problema para a metaheurística VNS implementada. A Figura 6b ilustra o vetor solução do mesmo exemplo citado no parágrafo anterior, este vetor tem um tamanho m , onde o m é a quantidade de vias da malha viária da instância

do problema. Cada posição do vetor refere-se a uma via (i, j) que pode receber um valor inteiro de 1 a 3 que designa o seguinte:

- Se o valor de uma posição do vetor solução for igual a 1, significa que a via (i, j) da malha viária que está sendo representada pela posição do vetor, é uma via de mão única no sentido de i para j .
- Se o valor de uma posição do vetor solução for igual a 2, significa que a via (i, j) da malha viária que está sendo representada pela posição do vetor, é uma via de mão única no sentido de j para i .
- Se o valor de uma posição do vetor solução for igual a 3, significa que a via (i, j) da malha viária que está sendo representada pela posição do vetor, é uma via de mão dupla no sentido de i para j e de j para i .

Na Figura 6b pode-se notar que o vetor tem apenas o valor 3 (três) em todas as suas posições, isso ocorre devido a instância de referência possuir apenas vias de mão dupla.

4.3.2 Perturbação

Como já mencionado neste trabalho na Seção 2.3.1, a metaheurística VNS tem um procedimento de perturbação que visa realizar algum tipo de modificação na solução atual para tentar explorar novas vizinhanças e também fugir de ótimos locais. Com isso, foi implementado uma perturbação que realiza modificações nas vias representada pelo vetor solução de forma aleatória.

No Algoritmo 4 é exposto o pseudocódigo da implementação do método da perturbação, onde tem como entrada uma solução (s), a quantidade de vias da instância (num_vias), e uma quantidade de modificações (qtd) a ser realizada pelo método. Dentro do laço de repetição do algoritmo será realizado um sorteio de um número aleatório que representa uma das posições do vetor solução (linha 4), após isso, será modificada a opção da via também de forma aleatória (representada por um valor de 1 a 3) da posição sorteada do vetor solução (linha 5), continuando a execução até o critério de parada do laço ser atingido.

4.3.3 Busca Local

Para a busca local da metaheurística VNS implementada, como já mencionado na Seção 2.3.1, foi proposto e utilizado o método de descida em vizinhança variável (*Variable Neighborhood Descent* - VND) (MLADENOVIĆ; HANSEN, 1999). Foi desenvolvido dois métodos de descida para o VND com o objetivo de explorar a vizinhança de s' (solução

Algoritmo 4: PERTUBAÇÃO

```

Entrada:  $s, num\_vias, qtd$ 
1 inicio
2    $cont \leftarrow 1$ 
3   enquanto  $cont \leq qtd$  faça
4      $sort \leftarrow rand(num\_vias)$  // sortear um número de 1 a  $num\_vias$ 
5      $s[sort] \leftarrow modificar\_via(s[sort])$  // sortear uma opção da via diferente
6      $cont \leftarrow cont + 1$ 
7   fim
8 fim
retorna :s

```

após o procedimento de perturbação) e encontrar um vizinho com o valor da função objetivo melhor que s' (primeira melhoria).

O método de descida 1 percorre todas as posições do vetor solução e realiza uma modificação em cada posição, avalia-se a solução e verifica se houve melhoria na função objetivo para poder a solução s' passar a ser o novo vizinho modificado, e continuar o procedimento. A modificação é efetuada da seguinte forma: (i) Se a posição do vetor solução está representando uma via de mão única, ele mudará para uma via de mão única no sentido inverso, ou seja, se o valor da posição é igual a 1, mudará para o valor 2, e se tem o valor 2, receberá o valor 1; (ii) Se a posição do vetor solução está representando uma via de mão dupla, ele mudará para uma via de mão única no sentido de i para j ou de j para i de forma aleatória, ou seja, de o valor da posição é igual a 3, receberá o valor 1 ou 2 de forma aleatória. O Algoritmo 5 apresenta o pseudocódigo do método de descida 1 que foi descrito.

Algoritmo 5: DESCIDA 1

```

Entrada:  $s', num\_vias$ 
1 inicio
2    $s'' \leftarrow s'$ 
3   para  $i = 1$  até  $num\_vias$  faça
4     se  $s'[i] = 1$  então
5        $s''[i] \leftarrow 2$ 
6     fim
7     se  $s'[i] = 2$  então
8        $s''[i] \leftarrow 1$ 
9     fim
10    senão
11       $s''[i] \leftarrow rand(1,2)$  // sortear aleatoriamente o valor 1 ou 2
12    fim
13    // Verificar se houve melhoria
14    se  $f(s'') < f(s')$  então
15       $s' \leftarrow s''$ 
16    fim
17    senão
18       $s'' \leftarrow s'$ 
19    fim
20 fim
retorna :s'

```

O método de descida 2 tem dois laços de repetição que executa um e depois o outro, e tem basicamente a mesma ideia do método de descida 1, mudando apenas como

é realizada a modificação de cada posição do vetor. No primeiro laço, a modificação é realizada da seguinte forma: (i) Se a posição do vetor solução tiver o valor 1 ou 3, a posição receberá o valor 2; (ii) Se a posição tiver o valor 2, ela mudará o seu valor para 3. Já segundo laço, se a posição do vetor solução tiver o valor 2 ou 3, a posição receberá o valor 1, ou se a posição tiver o valor 1, ela mudará o seu valor para 3. O Algoritmo 6 apresenta o pseudocódigo do método de descida 2 descrito anteriormente.

Algoritmo 6: DESCIDA 2

```

Entrada:  $s'$ ,  $num\_vias$ 
1 inicio
2    $s'' \leftarrow s'$ 
   // Primeiro laço de repetição
3   para  $i = 1$  até  $num\_vias$  faça
4     se  $s'[i] = 1$  ou  $s'[i] = 3$  então
5        $s''[i] \leftarrow 2$ 
6     fim
7     se  $s'[i] = 2$  então
8        $s''[i] \leftarrow 3$ 
9     fim
   // Verificar se houve melhoria
10    se  $f(s'') < f(s')$  então
11       $s' \leftarrow s''$ 
12    fim
13    senão
14       $s'' \leftarrow s'$ 
15    fim
16  fim
   // Segundo laço de repetição
17  para  $i = 1$  até  $num\_vias$  faça
18    se  $s'[i] = 2$  ou  $s'[i] = 3$  então
19       $s''[i] \leftarrow 1$ 
20    fim
21    se  $s'[i] = 1$  então
22       $s''[i] \leftarrow 3$ 
23    fim
   // Verificar se houve melhoria
24    se  $f(s'') < f(s')$  então
25       $s' \leftarrow s''$ 
26    fim
27    senão
28       $s'' \leftarrow s'$ 
29    fim
30  fim
31 fim
   retorna :  $s'$ 

```

4.3.4 Detalhamento da Implementação

Um dos objetos de estudo do presente trabalho foi o desenvolvimento da metaheurística VNS (MLADENOVIC; HANSEN, 1997) aplicado ao problema de projeto de rede. A implementação do VNS baseou-se na sua estrutura tradicional, contendo os procedimentos fundamentais exposto na Seção 2.3.1. No Algoritmo 7 é ilustrado o pseudocódigo de como ficou a implementação da metaheurística proposta, onde temos o VNS utilizando o VND como procedimento de busca local. O diferencial desta implementação, é a questão da quantidade de modificações do método de perturbação que pode ser incrementado até uma

quantidade máxima (linha 4 do algoritmo) quando não há melhoria na solução. Depois de alguns experimentos de testes durante o desenvolvimento, ficou definido que a quantidade máxima permitida de modificações da solução na perturbação seria 20 (vinte) por cento do número de vias da instância, e também, que a quantidade de modificações começa em 1 (linha 3), podendo ser incrementada depois de 50 iterações sem melhoria da solução atual (linha 16 a 22). Percebe-se que no momento em que é encontrado uma solução melhor depois da busca local (linha 9), é atualizada a solução atual (linha 10), a quantidade de modificações do método de perturbação recebe o valor 1 (linha 11), e também, o contador de iterações sem melhoria da solução recebe o valor 1 (linha 12).

Algoritmo 7: VNS-VND

```

Entrada: num_vias
1 inicio
2    $s \leftarrow \text{solucaoInicial}()$ 
3    $qtd \leftarrow 1$ 
4    $qtd\_max \leftarrow 0.2 * \text{num\_vias}$ 
5    $cont \leftarrow 1$ 
6   enquanto Critério de parada não satisfeito faça
7      $s' \leftarrow \text{pertubacao}(s, \text{num\_vias}, qtd)$  // (Algoritmo 4)
8      $s'' \leftarrow \text{buscaLocal}(s', \text{num\_vias})$  // VND com os métodos de descida (Algoritmo 5 e 6)
9     se  $f(s'') < f(s)$  então
10       $s \leftarrow s''$ 
11       $qtd \leftarrow 1$ 
12       $cont \leftarrow 1$ 
13     fim
14     senão
15        $cont \leftarrow cont + 1$ 
16       se  $cont = 50$  então
17          $qtd \leftarrow qtd + 1$ 
18         se  $qtd > qtd\_max$  então
19            $qtd \leftarrow 1$ 
20         fim
21        $cont \leftarrow 1$ 
22     fim
23   fim
24 fim
25 retorna :s // Melhor solução encontrada

```

Na avaliação de uma solução ($f(s)$), foi implementado a abordagem descrita na Seção 4.1, onde necessita da atualização do grafo (representação computacional da malha viária) a partir de uma solução, execução do algoritmo de Floyd-Warshall para encontrar os caminhos mais curtos de acordo com a demanda de viajantes da instância, e a utilização da equação da função objetivo do problema para obter o valor da solução ($f(s)$).

5 Experimentos e Resultados

Este capítulo tem como objetivo mostrar os experimentos computacionais realizados e seus respectivos resultados. Os experimentos computacionais foram feitos baseados no modelo matemático proposto implementado no CPLEX e na abordagem proposta mencionada no Capítulo 4. Além disso, são exibidas comparações entre os resultados da metaheurística proposta com os do modelo exato, além de alguns resultados disponíveis na literatura.

5.1 Instâncias do Problema

Os experimentos computacionais foram executados em duas classes de instâncias. A primeira classe C1 contém 3 (três) instâncias, onde uma foi criada originalmente no trabalho de Drezner e Wesolowsky (1997), a segunda foi gerada a partir de informações de uma instância deste mesmo trabalho (DREZNER; WESOLOWSKY, 1997), e a terceira instância é uma muito utilizada na literatura especializada para validar os métodos desenvolvidos que é a rede Sioux Falls. O segundo grupo C2 possui também 3 (três) instâncias, as quais foram criadas e utilizadas nos trabalhos: (DREZNER; WESOLOWSKY, 1997; DREZNER; SALHI, 2000; DREZNER; SALHI, 2002).

5.1.1 Classe C1 de Instâncias

A classe C1 tem 3 instâncias com os seguintes detalhes:

- (i) A primeira foi proposta por Drezner e Wesolowsky (1997) onde o grafo representando a malha viária contém 14 vértices e 20 arestas, e a demanda desta instância é igual a 1 para todos os pares de vértices diferentes.
- (ii) A segunda instância foi construída a partir de outra instância criada no trabalho de Drezner e Wesolowsky (1997) que fará parte da classe C2. A instância foi gerada da seguinte maneira: foi considerado apenas os vértices que estão em coordenadas x e y no intervalo $[1, 20]$ do Quadro 3, esse quadro contém coordenadas dos vértices para geração de instâncias utilizada no trabalho de Drezner e Wesolowsky (1997). Após isso, dois vértices são ligados por uma aresta se $\max\{|x_i - x_j|, |y_i - y_j|\} \leq 6$, ou seja, se o máximo entre as diferenças das coordenadas for menor ou igual a 6. Com isso, o grafo ficou com 18 vértices e 41 arestas. O custo associado a cada aresta é calculado pela distância euclidiana entre os vértices arredondado para o inteiro mais próximo. A demanda w_{ij} de uma origem i e um destino j desta instância foi

gerada para cada par de vértice pelo cálculo da distância euclidiana d_{ij} entre os dois, onde $w_{ij} = (100)/d_{ij}$ arredondado para o inteiro mais próximo.

- (iii) A terceira instância desta classe é a Sioux Falls, com 24 vértices e 38 arestas. Essa instância como já mencionado foi utilizada em diversos trabalhos na literatura especializada e as informações sobre esta instância está disponível em <<https://github.com/bstabler/TransportationNetworks/tree/master/SiouxFalls>>. Para os experimentos do presente trabalho, as informações consideradas desta instância foram os vértices, as ligações das arestas, os custos quando não há fluxo em cada aresta, e a demanda OD para a rede.

A Tabela 4 mostra a nomenclatura de cada instância desta classe e uma breve descrição.

Tabela 4 – Descrição das instâncias da classe C1.

Instância	Descrição
PQ_14_20	Rede pequena com 14 vértices e 20 arestas (vias)
PQ_18_41	Rede pequena com 18 vértices e 41 arestas (vias)
SF_21_38	Rede Sioux Falls com 21 vértices e 38 arestas (vias)

Fonte – Elaborado pelo autor.

5.1.2 Classe C2 de Instâncias

As instâncias deste grupo C2 foram geradas basicamente da mesma forma que a segunda instância da classe C1, mas com todos os vértices do Quadro 3, e se diferenciando apenas como dois vértices são conectados por uma aresta. Assim, as instâncias ficaram da seguinte forma:

- (i) A primeira instância tem 40 vértices e 65 arestas. Dois vértices foram ligados por uma aresta se $\max\{|x_i - x_j|, |y_i - y_j|\} \leq 5$, ou seja, se o máximo entre as diferenças das coordenadas dos vértices for menor ou igual a 5.
- (ii) A segunda instância tem 40 vértices e 99 arestas. Dois vértices foram ligados por uma aresta se $\max\{|x_i - x_j|, |y_i - y_j|\} \leq 6$, ou seja, se o máximo entre as diferenças das coordenadas dos vértices for menor ou igual a 6.
- (iii) A terceira instância tem 40 vértices e 164 arestas. Dois vértices foram ligados por uma aresta se $\max\{|x_i - x_j|, |y_i - y_j|\} \leq 8$, ou seja, se o máximo entre as diferenças das coordenadas dos vértices for menor ou igual a 8.

Na Tabela 5 é apresentado a nomenclatura de cada instância desta classe C2 e uma breve descrição.

Quadro 3 – Localização dos vértices

Vértice	x	y	Vértice	x	y
1	1	13	21	29	23
2	7	16	22	15	23
3	27	13	23	23	12
4	17	18	24	4	23
5	12	21	25	29	9
6	19	4	26	3	19
7	9	6	27	26	19
8	1	8	28	20	8
9	26	7	29	11	25
10	15	6	30	13	2
11	23	21	31	8	28
12	3	28	32	29	2
13	19	13	33	4	6
14	2	3	34	24	25
15	17	10	35	11	10
16	22	1	36	5	11
17	21	27	37	29	16
18	17	26	38	15	29
19	14	13	39	27	29
20	11	15	40	8	2

Fonte – Adaptado de Drezner e Wesolowsky (1997)

Tabela 5 – Descrição das instâncias da classe C2.

Instância	Descrição
MS_40_65	Rede média esparsa com 40 vértices e 65 arestas (vias)
ME_40_99	Rede média com 40 vértices e 99 arestas (vias)
MD_40_164	Rede média densa com 40 vértices e 164 arestas (vias)

Fonte – Elaborado pelo autor.

5.2 Modelo Matemático

Para validar a nova formulação matemática para o problema de projeto de rede proposta neste trabalho, foi implementado o modelo matemático no *software* CPLEX versão 12.8 na linguagem de programação C++ por meio da tecnologia *Concert Technology*¹. O CPLEX foi executado em um computador com o sistema operacional *Ubuntu* 18.04.3 LTS 64 bits, com 8 Gb de memória RAM e processador Intel Core i5-7200U de 2.5 Ghz. Notebook fabricado pela LENOVO de modelo: IdeaPad 320 80YH0006BR.

Os experimentos do modelo matemático no CPLEX foram realizado de três formas apenas nas instâncias da classe C1 devido serem instâncias menores e também pela

¹ *Concert Technology*: a tecnologia é um conjunto de bibliotecas que permite que um programador incorpore otimizadores CPLEX em aplicativos C++, Java ou .NET.

complexidade do problema. A primeira forma do experimento foi configurado no CPLEX com um tempo de limite máximo de um dia (equivalente a 24 horas ou 86400 segundos) de execução para poder encontrar a solução ótima. Nas outras 2 formas, o CPLEX foi configurado para parar de executar assim que encontrar uma solução inteira viável que esteja dentro de dez (10) e vinte (20) por cento do ideal (solução ótima), ou seja, com a tolerância do GAP de 10% e 20%. Nos próximos quadros, são apresentados os resultados obtidos para cada configuração do experimento. Cada quadro tem o valor da função objetivo da melhor solução (**#min**) possível do experimento e o tempo de execução ($\tau(s)$) gasto em segundos.

No Quadro 4 são mostrados os resultados para a primeira configuração do experimento (parâmetro de limite máximo de tempo de execução de um dia) para as instâncias da classe C1 com alfas de 0.5 a 0.8. Os valores da função objetivo que estão em destaque no quadro, são as soluções ótimas encontradas pelo CPLEX, e os outros que não estão em destaque são soluções viáveis que o CPLEX conseguiu obter com a execução até atingir o parâmetro de limite máximo de tempo. Com os resultados da instância PQ_14_20, a formulação matemática proposta para o problema de projeto de rede neste trabalho é válido, visto que foram encontrados os mesmos resultados obtidos pelo modelo matemático do trabalho de Drezner e Wesolowsky (1997) para esta mesma instância.

Quadro 4 – Resultados do CPLEX com o parâmetro de limite máximo de tempo de execução.

Instância	$\alpha = 0.5$		$\alpha = 0.6$		$\alpha = 0.7$		$\alpha = 0.8$	
	#min	$\tau(s)$	#min	$\tau(s)$	#min	$\tau(s)$	#min	$\tau(s)$
PQ_14_20	1385	1006,05	1656,6	1343,843233	1859,2	2424,30	1973	42,02
PQ_18_41	21001,5	86400	25299,8	86400	29188,7	86400	31035	31708,14
SF_21_38	2446000	86400	2863140	86400	3125550	86400	3170900	106,59

Fonte – Elaborado pelo autor.

Os resultados referentes aos experimentos com o parâmetro de tolerância do GAP de 20 e 10 por cento são apresentados nos Quadros 5 e 6, respectivamente. Nesses quadros, são exibidos os valores da função objetivo das soluções viáveis encontradas dentro da tolerância do GAP, e também o tempo computacional de execução (em segundos). Analisando os resultados desses quadros, pode-se notar que para as instâncias PQ_14_20 e SF_21_38 com o alfa igual a 0.8, mesmo com a definição da tolerância do GAP, foi encontrado a solução ótima, isso porque a solução viável inteira dentro da tolerância definida é a melhor solução, ou seja, a solução ótima.

Na Figura 7 é mostrado uma comparação dos resultados obtidos para as três configurações dos experimentos no CPLEX em relação a função objetivo para a instância PQ_14_20. Percebe-se que os melhores resultados são para a configuração do limite de tempo, isso porque foi definido um tempo máximo de um dia de execução, mas foi possível

Quadro 5 – Resultados do CPLEX com o parâmetro de tolerância do GAP = 20%.

Instância	$\alpha = 0.5$		$\alpha = 0.6$		$\alpha = 0.7$		$\alpha = 0.8$	
	#min	$\tau(s)$	#min	$\tau(s)$	#min	$\tau(s)$	#min	$\tau(s)$
PQ_14_20	1630	290,48	1875,4	266,63	2166,7	202,66	1973	29,17
PQ_18_41	22990	1405,51	26885	2477,13	31128	2129,74	31156,4	323,01
SF_21_38	2679250	4913	3135360	2186,56	3265610	1980,09	3170900	105,71

Fonte – Elaborado pelo autor.

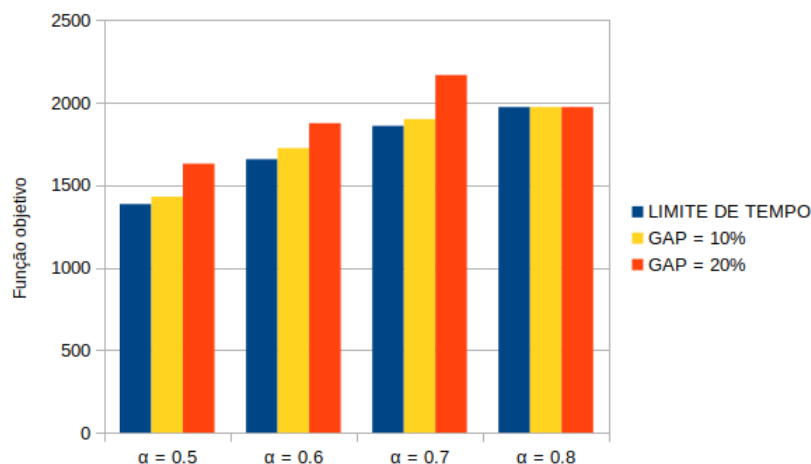
Quadro 6 – Resultados do CPLEX com o parâmetro de tolerância do GAP = 10%.

Instância	$\alpha = 0.5$		$\alpha = 0.6$		$\alpha = 0.7$		$\alpha = 0.8$	
	#min	$\tau(s)$	#min	$\tau(s)$	#min	$\tau(s)$	#min	$\tau(s)$
PQ_14_20	1430	291,92	1724,4	440,07	1900,4	270,66	1973	29,40
PQ_18_41	22261,5	1457,76	25999,2	4834,90	29829,8	5229,97	31156,4	323,68
SF_21_38	2476900	12245,49	2934800	5407,20	3265610	1972,49	3170900	106,10

Fonte – Elaborado pelo autor.

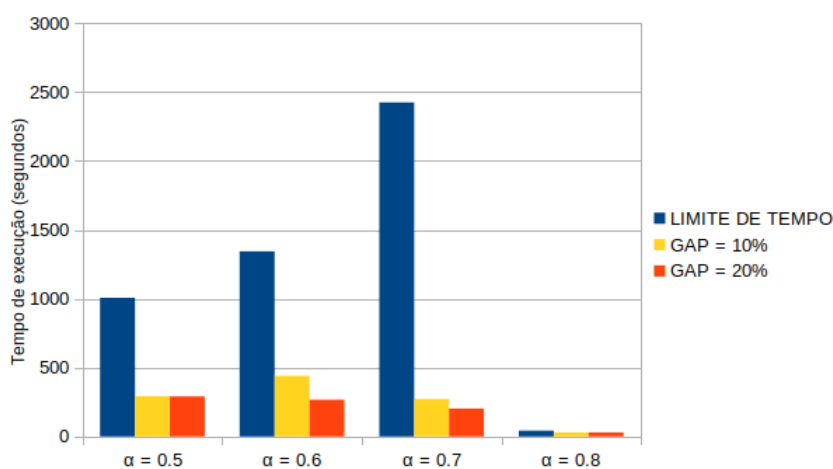
encontrar a solução ótima com um tempo bem menor sem atingir o limite de tempo. Note que, os resultados para o GAP com 10% e 20% na mesma figura são bem próximos da solução ótima. Já na Figura 8, temos a comparação do tempo computacional de execução gasto para os experimentos, e veja que para a configuração da tolerância do GAP obteve-se um tempo de execução bem menor do que para a configuração ajustando o limite máximo de tempo. Com isso, a definição da tolerância do GAP se torna um opção interessante para os experimentos computacionais a modelos matemáticos de problemas de otimização combinatória no CPLEX ou em algum outro *solver* matemático.

Figura 7 – Comparação dos resultados obtidos para as três configurações dos experimentos no CPLEX para a instância PQ_14_20.



Fonte – Elaborada pelo autor.

Figura 8 – Comparação do tempo de execução para as três configurações dos experimentos no CPLEX para a instância PQ_14_20.



Fonte – Elaborada pelo autor.

5.3 VNS

A metaheurística VNS proposta neste trabalho foi implementada na linguagem de programação C utilizando o editor de texto Sublime. O VNS foi executado em um terminal do mesmo computador que foi executado o CPLEX.

Os experimentos foram realizados em todas as instâncias das classes C1 e C2, sendo executado 30 (trinta) vezes para cada uma delas. O critério de parada utilizado pelo VNS foi definido depois de alguns experimentos de testes, que é executar uma quantidade máxima iterações igual a vinte e quatro (24) multiplicado pela quantidade de vias da malha viária da instância. O valor da solução inicial para cada instância nos experimentos é exposto na Tabela 6. A solução inicial como já mencionado neste trabalho, representa a malha viária quando todas as vias são vias de mão dupla.

Tabela 6 – Valor da solução inicial para cada instância.

Instância	Solução Inicial
PQ_14_20	1998
PQ_18_41	31632
PQ_21_38	3176000
MS_40_65	189036
ME_40_99	166870
MD_40_164	159006

Fonte – Elaborado pelo autor

Nos Quadros 7 e 8 são mostrados os resultados de forma resumida, os resultados dos experimentos com mais detalhes encontram-se no Apêndice A. Em cada quadro, é mostrado o valor da função objetivo da melhor solução (**#min**) encontrada das trinta execuções, a

média das soluções ($\bar{\chi}$), o número de vezes (φ) das execuções que foi encontrado a solução mínima (**#min**) e o tempo médio ($\overline{\tau(s)}$) das execuções em segundos.

Com os resultados apresentados do Quadro 7, podemos notar que para a instância PQ_14_20 a metaheurística VNS obteve o valor ótimo em todas suas execuções para todos os alfas (α), já que esses valores são iguais ao do modelo matemático apresentado no Quadro 4 para a mesma instância. Ainda observando o mesmo quadro, para os resultados das outras duas instâncias com alfa (α) igual a 8, o VNS conseguiu encontrar a solução ótima em todas as execuções. Para os demais resultados do mesmo quadro, a solução mínima (**#min**) obtida pelo VNS provavelmente também são soluções ótimas, visto que são resultados melhores que do método exato para todas as configurações.

Quadro 7 – Resultados gerais do VNS para as instâncias da classe C1.

Instância	$\alpha = 0.5$				$\alpha = 0.6$				$\alpha = 0.7$				$\alpha = 0.8$			
	#min	$\bar{\chi}$	φ	$\overline{\tau(s)}$	#min	$\bar{\chi}$	φ	$\overline{\tau(s)}$	#min	$\bar{\chi}$	φ	$\overline{\tau(s)}$	#min	$\bar{\chi}$	φ	$\overline{\tau(s)}$
PQ_14_20	1385	1385	30	0,86	1656,6	1656,6	30	0,93	1859,2	1859,2	30	1,12	1973	1973	30	1,17
PQ_18_41	20932	20981,9	12	7,69	25118,4	25169,1	11	8,55	29006,3	29034,65	23	9,70	31035	31035	30	11,49
SF_21_38	2439450	2442245	12	17,58	2845420	2846413,33	16	22,12	3123230	3123383	13	23,95	3170900	3170900	30	21,64

Fonte – Elaborado pelo autor.

No Quadro 8 são mostrados os resultados obtidos pelo VNS para as instâncias da classe C2. Para essas instâncias, não é conhecido o valor da solução ótima, apenas se sabe a melhor solução encontrada na literatura para cada uma das instâncias no trabalho de Drezner e Salhi (2002). As melhores soluções (**#min**) encontrada pelo VNS que não estão em destaque no quadro, são as mesmas conhecidas como melhor no trabalho de Drezner e Salhi (2002). Já as que estão em destaque na instância MD_40_164 são resultados melhores do que os já conhecido na literatura. Na Seção 5.4.2 é exposto as análises e comparações dos resultados do VNS com os resultados conhecido da literatura (trabalho de Drezner e Salhi (2002)).

Quadro 8 – Resultados gerais do VNS para as instâncias da classe C2.

Instância	$\alpha = 0.5$				$\alpha = 0.6$				$\alpha = 0.7$				$\alpha = 0.8$			
	#min	$\bar{\chi}$	φ	$\overline{\tau(s)}$	#min	$\bar{\chi}$	φ	$\overline{\tau(s)}$	#min	$\bar{\chi}$	φ	$\overline{\tau(s)}$	#min	$\bar{\chi}$	φ	$\overline{\tau(s)}$
MS_40_65	143259	143615,72	12	191,99	165481,2	165531,36	23	229,33	181355,7	181355,7	30	270,22	188621	188621	30	312,24
ME_40_99	108046	108244,78	2	470,57	127949,6	128138,18	5	524,67	146431,3	146620,70	15	589,42	160191,8	160314,41	13	688,42
MD_40_164	92104	92308,13	1	1330,15	110524,8	110738,16	1	1419,50	128935	129159,43	1	1674,43	144676	144794,95	5	2089,85

Fonte – Elaborado pelo auto.r

No Apêndice B são exibidos os grafos que representam a malha viária para cada instância da classe C1 e C2, todas estas com vias de mão dupla (solução inicial), e também, os melhores resultados encontrado pelo VNS para os alfas de 0.5 a 0.8, que representam a melhor configuração da malha com vias de mão única e mão dupla.

5.4 Análises dos Resultados

Nesta seção, iremos analisar e comparar os resultados do método de solução proposta desenvolvida. Na Subseção 5.4.1, é avaliado o desempenho do VNS para as instâncias da classe C1 em relação aos resultados obtidos pelo modelo matemático proposto implementado no CPLEX. Para os resultados alcançados pelo VNS para as instâncias C2, são analisados e comparados com resultados de quatro métodos desenvolvidos no trabalho de Drezner e Salhi (2002) na Subseção 5.4.2.

5.4.1 Comparação do VNS com o Método Exato

Nesta subseção, realiza-se comparações e análises dos resultados da metaheurística VNS com os do método exato para as instâncias da classe C1. A comparação é efetuada em relação ao valor da solução encontrada (**#min**) e o tempo computacional ($\tau(s)$) gasto para as três configurações dos experimentos do método exato, e para o VNS é levado em consideração a média das soluções ($\bar{\chi}$) e do tempo ($\overline{\tau(s)}$) gasto para as trinta execuções do experimento do mesmo.

A Tabela 7 mostra uma comparação dos resultados obtidos pelo VNS para as instâncias da classe C1, com os resultados do método exato para as três configurações do experimento que são: a tolerância do GAP igual a 20 e a 10 por cento e limite de tempo máximo de execução.

Tabela 7 – Comparação dos resultados obtidos pelo VNS com os do método exato.

α	GAP = 20%		GAP = 10%		LIMITE DE TEMPO		VNS	
	#min	$\tau(s)$	#min	$\tau(s)$	#min	$\tau(s)$	$\bar{\chi}$	$\overline{\tau(s)}$
Rede Pequena com 14 vértices e 20 arestas (PQ_14_20)								
0.8	1973	29,17	1973	29,4	1973	42,02	1973	1,17
0.7	2166,7	202,66	1900,4	270,66	1859,2	2424,3	1859,2	1,12
0.6	1875,4	266,63	1724,4	440,07	1656,6	1343,84	1656,6	0,93
0.5	1630	290,48	1430	291,92	1385	1006,05	1385	0,86
Rede Pequena com 18 vértices e 41 arestas (PQ_18_41)								
0.8	31156,4	323,01	31156,4	323,68	31035	31708,14	31035	11,49
0.7	31128	2129,74	29829,8	5229,97	29188,7	86400	29034,65	9,70
0.6	26885	2477,13	25999,2	4834,9	25299,8	86400	25169,06	8,55
0.5	22990	1405,51	22261,5	1457,76	21001,5	86400	20981,9	7,69
Rede Sioux Falls com 21 vértices e 38 arestas (SF_21_38)								
0.8	3170900	105,71	3170900	106,1	3170900	106,59	3170900	21,64
0.7	3265610	1980,09	3265610	1972,49	3125550	86400	3123383	23,95
0.6	3135360	2186,56	2934800	5407,2	2863140	86400	2846413,33	22,12
0.5	2679250	4913	2476900	12245,49	2446000	86400	2442245	17,58

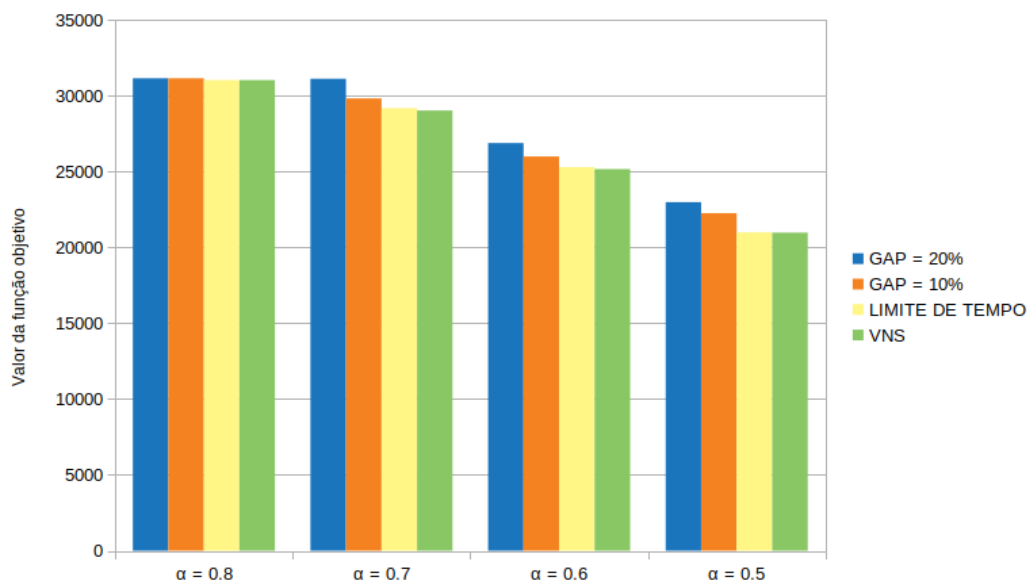
Fonte – Elaborado pelo autor.

Com os dados expostos na Tabela 7, nota-se que para a instância PQ_14_20 o VNS obteve como média das soluções os valores ótimos para todos os alfas, e o método

exato obteve o ótimo também com a configuração do limite de tempo máximo, mas gastou um tempo computacional maior que o do VNS. Ainda analisando os dados da mesma tabela, para as outras duas instâncias, podemos ver que o VNS tem os melhores resultados, tanto para a qualidade das soluções quanto para o tempo computacional gasto comparado com qualquer uma das formas de configuração do método exato, e até mesmo nos casos que o método exato atingiu o limite máximo de tempo (86400 segundos ou 1 dia) e não conseguiu alcançar a solução ótima, o VNS foi superior.

Nas Figuras 9 e 10 são mostrados comparações entre os resultados do método exato e da metaheurística VNS considerando o valor da função objetivo das soluções para as instâncias PQ_18_41 e SF_21_38, respectivamente. Nessas figuras é possível visualizar graficamente que os resultados do VNS é realmente superior aos do método exato. Não foram colocados os dados em relação ao tempo computacional gasto em um gráfico devido a uma discrepância enorme entre o método exato e o VNS, como podemos ver na Tabela 7. Por exemplo, o método exato com a tolerância do GAP igual a 20 por cento para as instâncias PQ_18_41 e SF_21_38, gastou mais de mil segundos para os alfas de 0.7 a 0.5, enquanto o VNS gastou no máximo 24 segundos.

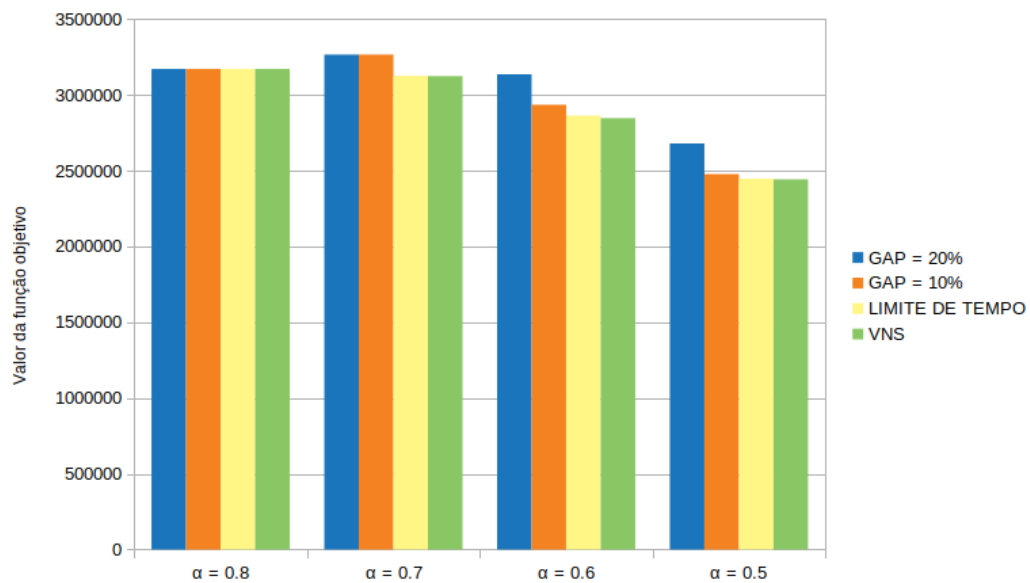
Figura 9 – Comparação das soluções do método exato e do VNS para a instância PQ_18_41.



Fonte – Elaborada pelo autor.

Com base nos gráficos e nas demais informações analisadas até aqui, é possível concluir que a metaheurística VNS desenvolvida tem um desempenho computacional formidável, e apresenta os melhores resultados para as instâncias da classe C1 do que as três formas de experimento do método exato.

Figura 10 – Comparação das soluções do método exato e do VNS para a instância SF_21_38.



Fonte – Elaborada pelo autor.

5.4.2 Comparação do VNS com Metaheurísticas da Literatura

Na subseção anterior, avaliou-se o desempenho do VNS em relação aos resultados do método exato para as instâncias C1. Como já mencionado, nesta Subseção será feita a comparação dos resultados do VNS com os disponíveis na literatura para as instâncias C2. Essa comparação é realizada em relação a melhor solução encontrada e a qualidade média das soluções. Em relação ao desempenho computacional, não foi possível comparar os resultados, pois há questões que dificultam as análises, como os equipamentos computacionais utilizados, linguagem de programação diferente, e a forma dos experimentos.

A Tabela 8 apresenta uma comparação dos resultados obtidos pelo VNS para as instâncias da classe C2, com os resultados dos métodos do trabalho de Drezner e Salhi (2002), que são: um método de descida (*DESCENT*), a metaheurística Busca Tabu (TABU), o *Simulated Annealing* (SA), e o Algoritmo Genético (AG).

Analisando os dados da Tabela 8, podemos ver que para a instância MS_40_65 para todos os alfas (0.5 a 0.8) apenas os algoritmos SA, AG e VNS conseguiram encontrar as melhores soluções conhecida, e na instância ME_40_99 só o AG e o VNS conseguiram obter as melhores soluções. Já para a instância MD_40_164, o VNS conseguiu alcançar melhores soluções (**#min**) do que os outros algoritmos para os alfas de 0.5 a 0.7, que até o momento desta pesquisa, as melhores soluções conhecidas eram do Algoritmo Genético (AG). Com isso, em relação a obter a melhor solução, verificamos que o VNS é superior em comparação aos outros algoritmos.

A Figura 11 apresenta uma comparação entre os algoritmos da literatura e o VNS

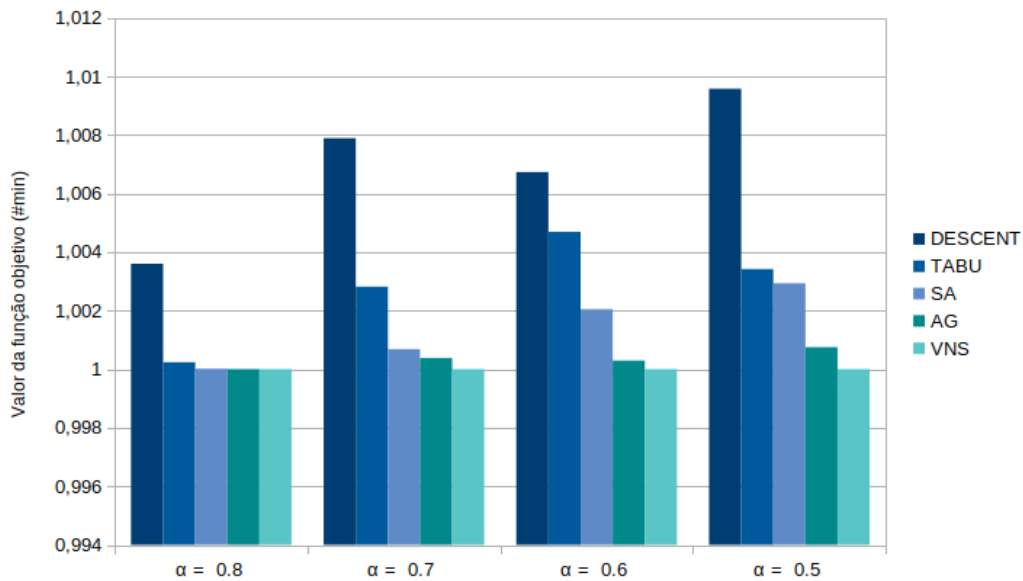
Tabela 8 – Comparação dos resultados obtidos pelo VNS com os da literatura.

α	DESCENT		TABU		SA		AG		VNS	
	#min	\bar{x}	#min	\bar{x}	#min	\bar{x}	#min	\bar{x}	#min	\bar{x}
Rede Média Esparso com 40 vértices e 65 arestas (MS_40_65)										
0.8	188621	188621	188621	188621	188621	188630	188621	188621	188621	188621
0.7	181994	183778	181355,7	182541	181355,7	181803	181355,7	181355,7	181355,7	181355,7
0.6	166926	170025	165595	167544	165481,2	167044	165481,2	165494	165481,2	165531,4
0.5	148004	152462	143591	145281	143259	145375	143259	143290	143259	143615,7
Rede Média com 40 vértices e 99 arestas (ME_40_99)										
0.8	160664	161157	160191,8	160772	160191,8	160499	160191,8	160209	160191,8	160314,4
0.7	147135	148876	146588	147928	146431,3	147217	146431,3	146484	146431,3	146620,7
0.6	129086	130390	128066	129134	128097	128741	127949,6	128019	127949,6	128138,2
0.5	108813	110152	108151	109119	108126	108927	108046	108123	108046	108244,8
Rede Média Densa com 40 vértices e 164 arestas (MD_40_164)										
0.8	145197	145694	144710	145276	144678	144923	144676	144709	144676	144794,9
0.7	129952	130587	129298	129835	129023	129517	128984	129172	128935	129159,4
0.6	111269	112116	111043	111540	110751	111227	110557	110685	110524,8	110738,2
0.5	92986	93485	92419	92904	92374	92725	92173	92271	92104	92308,1

Fonte – Elaborado pelo autor.

desenvolvido, considerando as melhores soluções (**#min**) encontradas para a instância MD_40_164. Na figura, podemos visualizar que realmente o VNS se sobressai em relação aos outros. Os dados representados no gráfico foram submetidos a uma normalização, sendo os valores da função objetivo de cada algoritmo normalizado pelos valores da função objetivo do VNS, já que agora são os melhores valores conhecido para a instância, como é possível ver na Tabela 8.

Figura 11 – Comparação das melhores soluções dos algoritmos da literatura e do VNS para a instância MD_40_164.

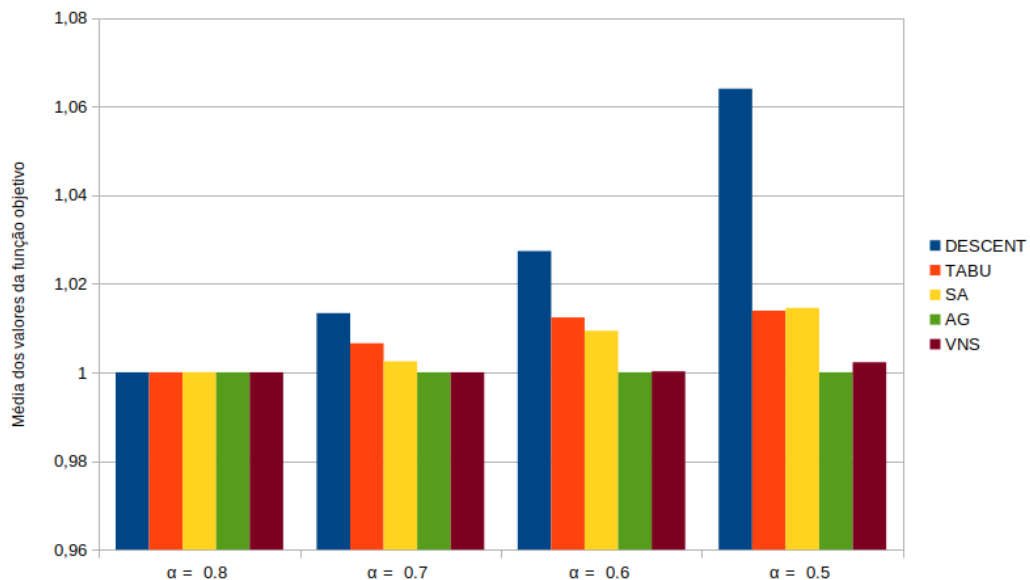


Fonte – Elaborada pelo autor.

Considerando os resultados da Tabela 8 em relação a média das soluções (\bar{x}) dos algoritmos, podemos notar que basicamente o AG tem as melhores médias para todas as

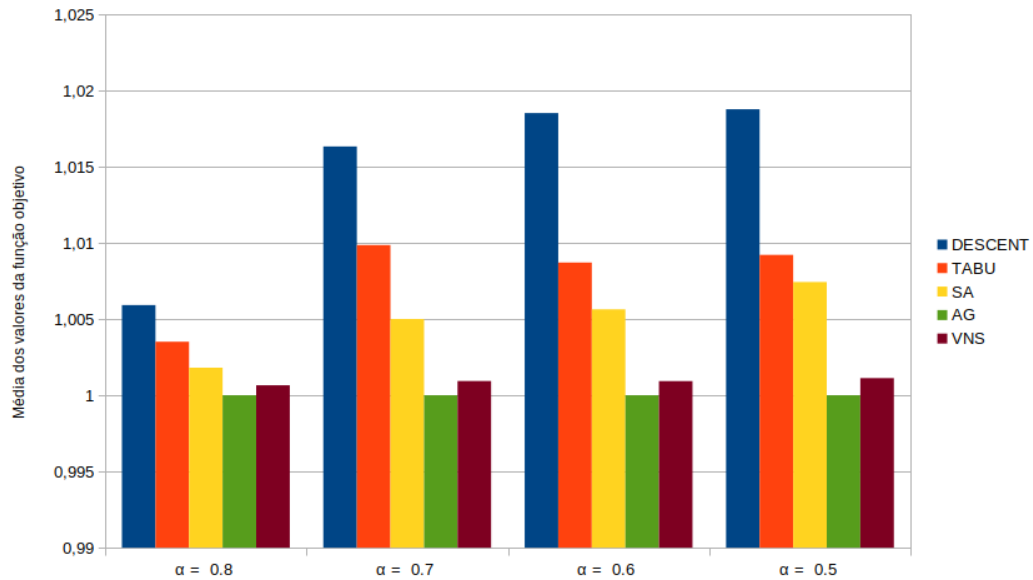
três instâncias com todos os alfas. O VNS fica atrás apenas do AG, e apresenta melhores médias que os demais algoritmos. Nas Figuras 12, 13 e 14 é possível visualizar graficamente o que foi dito anteriormente para as instâncias MS_40_65, ME_40_99 e MD_40_164, respectivamente. Nos gráficos das figuras, os dados passaram por uma normalização para facilitar a visualização e o entendimento, sendo que os valores médios da função objetivo de cada algoritmo foram normalizados pela melhor média entre os algoritmos em cada instância e para cada alfa, ou seja, todos os dados foram normalizados pelas médias do AG, exceto na instância MD_40_164 para o alfa igual a 0.7 onde os dados foram normalizados pela média do VNS, visto que foi a melhor entre todos os algoritmos, como mostra a Tabela 8.

Figura 12 – Comparação das médias das soluções dos algoritmos da literatura e do VNS para a instância MS_40_65.



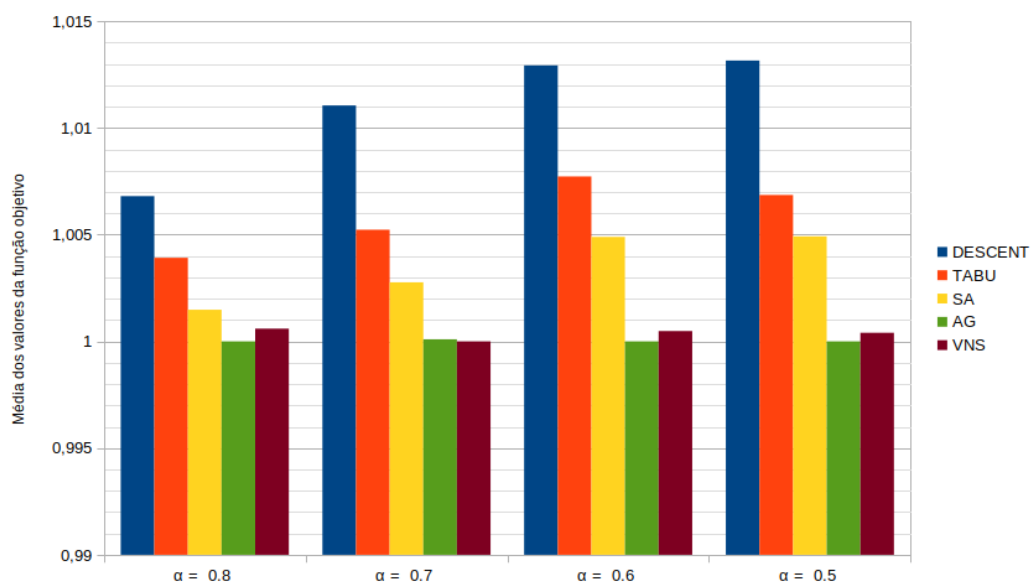
Fonte – Elaborada pelo autor.

Figura 13 – Comparação das médias das soluções dos algoritmos da literatura e do VNS para a instância ME_40_99.



Fonte – Elaborada pelo autor.

Figura 14 – Comparação das médias das soluções dos algoritmos da literatura e do VNS para a instância MD_40_164.



Fonte – Elaborada pelo autor.

6 Conclusões e Trabalhos Futuros

Neste capítulo é exibido as conclusões e os trabalhos futuros da presente pesquisa. A Seção 6.1 mostra as considerações acerca do trabalho desenvolvido. A ampliação desta pesquisa como trabalhos futuros são discutidos na Seção 6.2.

6.1 Conclusões

Neste trabalho foram exposto dois objetos de estudo: (i) o desenvolvimento de uma nova formulação matemática para o problema de projeto de rede com decisões discretas para determinar uma configuração ótima de uma malha viária com vias de mão única e mão dupla; e (ii) a implementação da metaheurística VNS para a resolução do problema.

No primeiro objeto de estudo foi realizado uma redução do modelo matemático de Drezner e Wesolowsky (1997) que utiliza três equações matemática para a restrição de conservação de fluxo, e o novo modelo proposto no presente trabalho usa apenas uma equação matemática para a mesma restrição, tornando-o mais simples e compacto. O modelo proposto, foi validado com a implementação do mesmo no CPLEX, e obtendo os mesmos resultados do modelo de Drezner e Wesolowsky (1997) para a instância PQ_14_20 (Rede Pequena com 14 vértices e 20 arestas).

O segundo objeto de estudo foi a implementação da metaheurística VNS como procedimento de solução para o problema de projeto de rede. No procedimento de solução para o problema houve também a implementação do algoritmo de Floyd-Warshall para encontrar os caminhos mais curtos da demanda entre origem e destino na malha viária. O VNS ficou responsável pela escolha dos projetos para modificação na malha viária, e o mesmo, utilizou como procedimento de busca local o VND (*Variable Neighborhood Descent*) proposto por Mladenović e Hansen (1999).

Para o novo modelo matemático proposto e a metaheurística VNS foram realizados experimentos computacionais para as classes de instâncias C1 e C2, onde o modelo implementado no CPLEX foi executado apenas nas instâncias da classe C1, e o VNS para todas as instâncias C1 e C2. Os resultados computacionais obtidos pelo VNS foram avaliados em relação a qualidade da solução, médias de tempo e valor médio das soluções durante as 30 execuções de cada uma das instâncias.

A comparação dos resultados entre o modelo matemático e o VNS para as instâncias C1, comprovou que a metaheurística VNS tem um desempenho superior em relação ao modelo exato, encontrando resultados melhores com pouco tempo de execução. Enquanto isso, na maioria dos casos o modelo exato necessitou de um tempo alto de execução para

obter a solução ótima ou próximo do ótimo, como foi visto nos experimentos computacionais. Desse modo, confirma-se que a utilização de metaheurísticas para problemas de otimização combinatória, em especial para o NDP, é um dos meios eficientes de resolução do problema.

Nos resultados obtidos pelo VNS para as instâncias C2, foram realizadas comparações com quatro métodos discutido no trabalho de Drezner e Salhi (2002), e o VNS mostrou que em termos de obter a melhor solução entre os algoritmos, ele foi superior. Já em relação a qualidade média das soluções, o VNS ficou depois do AG que tem as melhores médias para as instâncias C2.

Dessa forma, podemos concluir que a metaheurística VNS demonstrou ser um algoritmo promissor, conseguindo alcançar soluções excelentes, e também, a melhor solução conhecida na literatura com o tempo computacional razoável.

Como contribuições, este trabalho apresenta um novo modelo matemático para o problema de projeto de rede com decisões discretas para determinar uma configuração ótima de uma malha viária com vias de mão única e mão dupla, o desenvolvimento de uma metaheurística que ainda não tinha sido aplicado ao problema, e também, novos resultados melhores que o VNS conseguiu obter dos quais já conhecido para a instância (Rede Média Densa com 40 vértices e 164 arestas MD_40_164) na literatura.

6.2 Trabalhos Futuros

Como trabalho futuro, será realizado um estudo de caso em alguma cidade com a finalidade de gerar instâncias reais e tentar melhorar o trânsito de uma malha viária encontrando a melhor configuração da mesma com vias de mão única de mão dupla.

Tem-se por objetivo também o desenvolvimento de um estudo comparativo entre metaheurísticas populacionais e de trajetória para este problema, com a realização de análises e comparações para verificar qual das duas metodologias se adéqua melhor ao problema.

Outra possibilidade de trabalho é utilização de outro algoritmo para encontrar os caminhos mais curtos na abordagem de procedimento de solução deste trabalho, e também na adaptação de outros procedimentos de busca local e perturbação para o VNS.

Referências

ARENALES, M. *et al.* *Pesquisa Operacional*. Rio de Janeiro: Elsevier, 2007. ISBN 978-85-352-5193-7. Citado 3 vezes nas páginas 22, 25 e 27.

AUSIELLO, G. *et al.* *Complexity and Approximation: Combinatorial Optimization Problems and Their Approximability Properties*. Berlin: Springer, 2003. ISBN 3-540-65431-3. Citado na página 22.

BARÃO, F. R.; KRIPKA, M.; KRIPKA, R. M. L. Modelagem matemática para coleta seletiva no município de passo fundo - rs. In: LOPES, H. S.; RODRIGUES, L. C. de A.; STEINER, M. T. A. (Org.). *Meta-heurísticas em pesquisa operacional*. Curitiba, PR: Omnipax, 2013. p. 237–251. Citado na página 20.

BELLMAN, R. On a routing problem. *Quarterly of Applied Mathematics*, v. 16, n. 1, p. 87–90, Abril 1958. Citado na página 25.

BEN-AKIVA, M. *et al.* Dynamit: a simulation-based system for traffic prediction. *DACCORD Short Term Forecasting Workshop*, 1998. Citado na página 17.

BLUM, C.; ROLI, A. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Computing Surveys (CSUR)*, v. 35, n. 3, p. 268–308, 2003. Citado 3 vezes nas páginas 28, 29 e 30.

BROWNLEE, J. *Clever Algorithms. Nature-Inspired Programming Recipes*. 1st. ed. [S.l.]: LuLu, 2011. ISBN 978-1-4467-8506-5. Citado 2 vezes nas páginas 27 e 29.

CARVALHO, C. H. R.; PEREIRA, R. H. M. *Efeitos da variação da tarifa e da renda da população sobre a demanda de transporte público coletivo urbano no Brasil*. Brasília: [s.n.], 2011. IPEA. Citado na página 17.

CHOPARD, B.; TOMASSINI, M. *An Introduction to Metaheuristics for Optimization*. 1. ed. [S.l.]: Springer, 2018. ISBN 978-3-319-93072-5. Citado na página 27.

CHUZHOUY, J. *et al.* On the approximability of some network design problems. *ACM Transactions on Algorithms (TALG)*, v. 4, n. 2, 2008. Citado na página 32.

CNT; NTU. *Pesquisa mobilidade da população urbana 2017*. 2017. Confederação Nacional do Transporte, Associação Nacional das Empresas de Transportes Urbano. Citado na página 17.

CORMEN, T. H. *et al.* *Algoritmos - Teoria e Prática*. 3rd. ed. [S.l.]: Elsevier, 2012. ISBN 8535236996. Citado 3 vezes nas páginas 25, 26 e 27.

CUNHA, V. A.-M. C. da *et al.* Aplicação de um algoritmo grasp reativo para a alocação e sequenciamento de unidades de resgate. In: SIMPÓSIO BRASILEIRO DE PESQUISA OPERACIONAL, 50., 2018, Rio de Janeiro. *Anais eletrônicos...* Campinas, GALOÁ, 2018. Disponível em: <<https://proceedings.science/sbpo/papers/aplicacao-de-um-algoritmo-grasp-reativo-para-a-alocacao-e-sequenciamento-de-unidades-de-resgate>>. Acesso em: 06 fev. 2019. Citado na página 20.

- DESROCHERS, M.; SOUMIS, F. A generalized permanent labeling algorithm for the shortest path problem with time windows. *Information Systems Research - ISR*, v. 26, p. 191–212, 01 1988. Citado na página 24.
- DIJKSTRA, E. W. A note on two problems in connexion with graphs. *Numerische Mathematik*, v. 1, n. 1, p. 269–271, 1959. Citado na página 25.
- DORIGO, M. *Optimization, Learning and Natural Algorithms*. Tese (Doutorado) — Politecnico di Milano, Italy, 1992. Citado na página 29.
- DREZNER, Z.; SALHI, S. Selecting a good configuration of one-way and two-way routes using tabu search. *Control Cybernet*, v. 29, n. 3, p. 725–740, 2000. Citado 3 vezes nas páginas 18, 36 e 46.
- DREZNER, Z.; SALHI, S. Using hybrid metaheuristics for the one-way and two-way network design problem. *Naval Research Logistics (NRL)*, v. 49, p. 449–463, Agosto 2002. Citado 7 vezes nas páginas 18, 36, 46, 52, 53, 55 e 60.
- DREZNER, Z.; WESOLOWSKY, G. O. Selecting an optimum configuration of one-way and two-way routes. *Transportation Science*, v. 31, n. 4, p. 386–394, 1997. Citado 11 vezes nas páginas 18, 19, 33, 34, 35, 36, 39, 46, 48, 49 e 59.
- DUARTE, D. C. S. *LIPSTUD – Um método de otimização de fluxo de tráfego urbano baseado em proibição e permissão de conversões*. Dissertação (Mestrado) — Instituto de Informática da Universidade Federal de Goiás, Curso de Pós-Graduação em Computação, Goiânia/GO, 2012. Citado 5 vezes nas páginas 17, 18, 20, 32 e 37.
- FARAHANI, R. Z. *et al.* A review of urban transportation network design problems. *European Journal of Operational Research*, v. 229, n. 2, p. 281–302, 2013. Citado na página 18.
- FELLENDORF, M. Vissim: A microscopic simulation tool to evaluate actuated signal control including bus priority. *64th ITE Annual Meeting*, 1994. Citado na página 17.
- FEO, T. A.; RESENDE, M. G. C. Greedy randomized adaptive search procedures. *Journal of Global Optimization*, v. 6, n. 2, p. 109–133, 1995. Citado na página 29.
- FLOYD, R. W. Algorithm 97: Shortest path. *Communications of the ACM*, v. 5, n. 6, p. 345, 1962. Citado na página 25.
- FORD Jr., L. R. *Network Flow Theory*. Santa Monica, California: The RAND Corporation, 1956. (Papers P-923). Citado na página 25.
- FOULDS, L. R. A multi-commodity flow network design problem. *Transportation Research Part B: Methodological*, v. 15, n. 4, p. 273–283, 1981. Citado 2 vezes nas páginas 17 e 36.
- GAO, Z.; WU, J.; SUN, H. Solution algorithm for the bi-level discrete network design problem. *Transportation Research Part B: Methodological*, v. 36, n. 6, p. 479–495, 2005. Citado na página 32.
- GARCÍA-LÓPEZ, F. *et al.* The parallel variable neighborhood search for the p-median problem. *Journal of Heuristics*, v. 8, n. 3, p. 375–388, 2002. Citado na página 30.

GARFINKEL, R.; FERNÁNDEZ, E.; LOWE, T. J. The k-centrum shortest path problem. *TOP*, v. 14, n. 2, p. 279–292, Dezembro 2006. Citado na página 24.

GASPERO, L. D. *et al.* A hybrid ls-cp solver for the shifts and breaks design problem. In: BLESA, M. J. *et al.* (Org.). *Hybrid Metaheuristics*. [S.l.]: Springer, 2010. p. 46–61. Citado na página 28.

GLOVER, F. Future paths for integer programming and links to artificial intelligence. *Computers and Operations Research*, v. 13, n. 5, p. 533–549, 1986. Citado na página 29.

GLOVER, F.; KOCHENBERGER, G. A. *Handbook of Metaheuristics*. 1. ed. [S.l.]: Springer, 2003. ISBN 1-4020-7263-5. Citado na página 27.

GOLDBARG, M. C.; LUNA, H. P. L. *Otimização combinatória e programação linear: modelos e algoritmos*. Rio de Janeiro: Elsevier, 2005. ISBN 978-85-352-1520-5. Citado na página 24.

GOLDBERG, D. E. *Genetic Algorithms in Search, Optimization and Machine Learning*. Boston, MA: Addison-Wesley, 1989. (1). ISBN 0201157675. Citado na página 29.

HANSEN, P.; MLADENOVIĆ, N. Variable neighborhood search. In: PARDALOS, P. M.; RESENDE, M. G. C. (Ed.). *Handbook of Applied Optimization*. New York: Oxford University Press, 2002. Citado na página 30.

HANSEN, P.; MLADENOVIĆ, N. A tutorial on variable neighborhood search. *Le cahiers du GERARD*, n. 46, 2003. Citado na página 30.

HART, P. E.; NILSSON, N. J.; RAPHAEL, B. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, v. 4, n. 2, p. 100–107, 1968. Citado na página 25.

HÉLIODORE, F. *et al.* *Metaheuristics for Intelligent Electrical Networks*. 1. ed. [S.l.]: ISTE, 2017. v. 10. ISBN 978-1-84821-809-3. Citado 2 vezes nas páginas 28 e 29.

HOLLAND, J. H. *Adaptation in Natural and Artificial Systems*. Ann Arbor, MI: University of Michigan Press, 1975. Citado na página 29.

INGERMAN, P. Z. Algorithm 141: Path matrix. *Communications of the ACM*, v. 5, n. 11, p. 556, 1962. Citado na página 25.

JAYAKRISHNAN, R.; MAHMASSANI, H. S.; HU, T.-Y. An evaluation tool for advanced traffic information and management systems in urban networks. *Transportation Research Part C: Emerging Technologies*, v. 2, n. 3, p. 129–147, 1994. Citado na página 17.

KENNEDY, J.; EBERHART, R. C. Particle swarm optimization. In: *Proceedings of the IEEE International Conference on Neural Networks*. [S.l.: s.n.], 1995. p. 1942–1948. Citado na página 29.

KIRKPATRICK, S.; GELATT, C. D.; VECCHI, M. P. Optimization by simulated annealing. *Science*, v. 220, n. 4598, p. 671–680, 1983. Citado na página 29.

LACORTT, M.; KRIPKA, M.; KRIPKA, R. M. L. Modelos matemáticos para otimização do tráfego urbano semaforizado. *Tendências em Matemática Aplicada e Computacional*, v. 4, n. 3, p. 359–372, 2013. Citado na página 17.

- LEBLANC, L. J. An algorithm for the discrete network design problem. *Transportation Science*, v. 9, n. 3, p. 183–199, 1975. Citado 3 vezes nas páginas 17, 32 e 36.
- LEE, J. *A First Course in Combinatorial Optimization*. [S.l.]: Cambridge University Press, 2004. Citado na página 22.
- LONG, J. *et al.* A turning restriction design problem in urban road networks. *European Journal of Operational Research*, v. 206, n. 3, p. 569–578, 2010. Citado 2 vezes nas páginas 17 e 36.
- LOUI, R. P. Optimal paths in graphs with stochastic or multidimensional weights. *ACM*, v. 26, n. 9, p. 670–676, Setembro 1983. Citado na página 24.
- LOURENÇO, H.; MARTIN, O.; STÜTZLE, T. A beginner’s introduction to iterated local search. In: *Proceeding of the 4th Metaheuristics International Conference*. [S.l.: s.n.], 2001. p. 1–11. Citado na página 29.
- LOZANO, M.; BLUM, C. A hybrid metaheuristic for the longest common subsequence problem. In: BLESA, M. J. *et al.* (Org.). *Hybrid Metaheuristics*. [S.l.]: Springer, 2010. p. 1–15. Citado na página 28.
- MLADENOVIĆ, N.; HANSEN, P. Variable neighborhood search. *Computers Operations Research*, v. 24, n. 11, p. 1097–1100, 1997. Citado 3 vezes nas páginas 19, 29 e 44.
- MLADENOVIĆ, N.; HANSEN, P. Variable neighborhood search: Methods and recent applications. *Third Metaheuristics International Conference*, p. 275–280, 1999. Citado 4 vezes nas páginas 30, 31, 42 e 59.
- MOORE, E. F. The shortest path through a maze. In: *Proceedings of an International Symposium on the Theory of Switching, Cybernetics and Informatics, 1957, Part II*. Cambridge, Massachusetts: Harvard University Press, 1959. p. 285–292. Citado na página 25.
- MOSCATO, P. On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms. *Technical Report Caltech Concurrent Computation Program, C3P Report 826*, Pasadena, California, USA, 1989. Citado na página 29.
- ORDEN, A. The transshipment problem. *Management Science*, v. 2, n. 3, p. 276–285, Abril 1956. Citado na página 24.
- POLACEK, M. *et al.* A variable neighborhood search for the multi depot vehicle routing problem with time windows. *Journal of Heuristics*, v. 10, n. 6, p. 613–627, 2004. Citado na página 30.
- RESENDE, M. G. de C.; SILVA, R. M. de A. Grasp: Procedimentos de busca gulosos, aleatórios e adaptativos. In: LOPES, H. S.; RODRIGUES, L. C. de A.; STEINER, M. T. A. (Org.). *Meta-heurísticas em pesquisa operacional*. Curitiba, PR: Omnipax, 2013. p. 1–20. Citado 2 vezes nas páginas 22 e 23.
- RODRIGUES, B. da C.; SANTOS, A. G. dos. Alocação de motoristas para um conjunto de viagens utilizando heurística. In: SIMPÓSIO BRASILEIRO DE PESQUISA OPERACIONAL, 50., 2018, Rio de Janeiro. *Anais eletrônicos...* Campinas, GALOÁ, 2018. Disponível em: <<https://proceedings.science/sbpo/papers/>>

- alocacao-de-motoristas-para-um-conjunto-de-viagens-utilizando-heuristica>. Acesso em: 06 fev. 2019. Citado na página 20.
- STORN, R.; PRICE, K. *Differential Evolution - a simple and efficient adaptive scheme for global optimization over continuous spaces*. Berkeley, CA, 1995. Citado na página 29.
- STÜTZLE, T. G. *Local Search Algorithms for Combinatorial Problems: Analysis, Improvements, and New Applications*. Tese (Doutorado) — Darmstadt University of Technology, Italy, 1998. Citado na página 29.
- VOUDOURIS, C.; TSANG, E. P. K. Partial constraint satisfaction problems and guided local search. In: *Proceedings of Second International Conference on Practical Application of Constraint Technology (PACT'96)*. [S.l.: s.n.], 1996. p. 337–356. Citado na página 29.
- WARDROP, J. Some theoretical aspects of road traffic research. *Proceedings of the Institution of Civil Engineers, part II*, v. 1, n. 36, p. 325–378, 1952. Citado na página 33.
- WARSHALL, S. A theorem on boolean matrices. *Journal of the ACM (JACM)*, v. 9, n. 1, p. 11–12, 1962. Citado na página 25.
- YU, G.; YANG, J. On the robust shortest path problem. *Computers Operations Research*, v. 25, n. 6, p. 457–468, Junho 1998. Citado na página 24.
- ZHANG, H.; GAO, Z. Two-way road network design problem with variable lanes. *Journal of Science and Systems Engineering*, v. 16, n. 1, p. 50–61, 2007. Citado na página 33.
- ZHANG, Y.; CHEN, X. D. An optimization model for the vehicle routing problem in multiproduct frozen food delivery. *Journal of Applied Research and Technology*, v. 12, n. 2, p. 239–250, Abril 2014. Citado na página 20.

Apêndices

APÊNDICE A – Resultados Experimentais do VNS-VND

Quadro 9 – Resultados da instância PQ_14_20 da classe C1.

Execução	$\alpha = 0.5$		$\alpha = 0.6$		$\alpha = 0.7$		$\alpha = 0.8$	
	#min	$\tau(s)$	#min	$\tau(s)$	#min	$\tau(s)$	#min	$\tau(s)$
1	1385	0,864462	1656,6	0,952345	1859,2	1,174008	1973	1,154372
2	1385	0,830422	1656,6	0,948834	1859,2	1,150633	1973	1,180691
3	1385	0,858784	1656,6	0,874898	1859,2	1,174474	1973	1,170721
4	1385	0,86681	1656,6	0,910058	1859,2	1,063608	1973	1,183538
5	1385	0,835156	1656,6	0,920972	1859,2	1,162973	1973	1,188052
6	1385	0,854968	1656,6	0,954909	1859,2	1,169689	1973	1,17634
7	1385	0,836573	1656,6	0,937533	1859,2	1,116768	1973	1,172067
8	1385	0,842171	1656,6	0,94482	1859,2	1,072105	1973	1,167452
9	1385	0,862516	1656,6	0,925596	1859,2	1,067258	1973	1,16197
10	1385	0,839388	1656,6	0,917785	1859,2	1,09777	1973	1,134272
11	1385	0,835694	1656,6	0,915278	1859,2	1,09394	1973	1,157285
12	1385	0,848361	1656,6	0,930548	1859,2	1,152127	1973	1,17035
13	1385	0,860336	1656,6	0,939051	1859,2	1,082543	1973	1,164343
14	1385	0,841455	1656,6	0,92448	1859,2	1,153799	1973	1,178714
15	1385	0,832744	1656,6	0,949612	1859,2	1,167069	1973	1,151008
16	1385	0,83051	1656,6	0,909134	1859,2	1,145356	1973	1,187054
17	1385	0,857641	1656,6	0,914918	1859,2	1,167235	1973	1,178845
18	1385	0,839356	1656,6	0,923882	1859,2	1,185516	1973	1,197629
19	1385	0,859208	1656,6	0,955194	1859,2	1,114777	1973	1,165528
20	1385	0,817934	1656,6	0,951047	1859,2	1,176264	1973	1,186503
21	1385	0,870338	1656,6	0,896083	1859,2	1,082376	1973	1,183549
22	1385	0,890576	1656,6	0,906296	1859,2	1,145741	1973	1,154532
23	1385	0,860253	1656,6	0,92014	1859,2	1,077107	1973	1,154913
24	1385	0,974331	1656,6	0,953882	1859,2	1,107788	1973	1,199324
25	1385	0,849215	1656,6	0,919166	1859,2	1,110542	1973	1,203483
26	1385	0,843145	1656,6	0,942185	1859,2	1,155228	1973	1,172265
27	1385	0,87289	1656,6	0,925578	1859,2	1,080441	1973	1,198813
28	1385	0,906856	1656,6	0,956569	1859,2	1,090473	1973	1,173193
29	1385	0,913661	1656,6	0,92092	1859,2	1,122178	1973	1,186349
30	1385	0,878957	1656,6	0,924285	1859,2	1,058639	1973	1,16089
Médias	$\bar{\chi}$	$\bar{\tau}(s)$	$\bar{\chi}$	$\bar{\tau}(s)$	$\bar{\chi}$	$\bar{\tau}(s)$	$\bar{\chi}$	$\bar{\tau}(s)$
	1385	0,859157	1656,6	0,928867	1859,2	1,123948	1973	1,173801

Quadro 10 – Resultados da instância PQ_18_41 da classe C1.

Execução	$\alpha = 0.5$		$\alpha = 0.6$		$\alpha = 0.7$		$\alpha = 0.8$	
	#min	$\tau(s)$	#min	$\tau(s)$	#min	$\tau(s)$	#min	$\tau(s)$
1	20932	7,618735	25177,4	8,581258	29115,9	9,704543	31035	11,419655
2	20932	7,767775	25118,4	8,6893	29006,3	9,814211	31035	11,68647
3	20995,5	7,594849	25236,6	8,618823	29006,3	9,83286	31035	11,260193
4	20995,5	7,679008	25118,4	8,731572	29006,3	9,716938	31035	11,314644
5	20932	7,505133	25177,4	8,524594	29094,9	9,978267	31035	11,381067
6	20995,5	7,828505	25177,4	8,476493	29006,3	9,578822	31035	11,32537
7	20932	7,799716	25177,4	8,335423	29006,3	9,779406	31035	11,646563
8	20996	7,750205	25118,4	8,616391	29006,3	9,911506	31035	11,32686
9	20996	7,458455	25177,4	8,589503	29006,3	9,4635	31035	11,586751
10	21010	7,457542	25177,4	8,441389	29006,3	9,592718	31035	11,54895
11	20932	7,727136	25118,4	8,745743	29006,3	9,614284	31035	11,228007
12	20995,5	7,579108	25177,4	8,521408	29130,5	9,631181	31035	11,280928
13	20932	7,631369	25236,6	8,291488	29006,3	9,523438	31035	11,156018
14	21030,5	7,618609	25177,4	8,313285	29141,4	9,71466	31035	11,68435
15	20932	7,591902	25177,4	8,373252	29006,3	9,563797	31035	11,260424
16	21067	7,960991	25236,6	8,580026	29006,3	9,700042	31035	11,792276
17	20995,5	7,665072	25236,6	8,510693	29006,3	9,590004	31035	11,263505
18	20995,5	7,681563	25177,4	8,37381	29006,3	9,988768	31035	11,633069
19	20996	7,586301	25177,4	8,624841	29130,5	9,660919	31035	11,821952
20	20932	7,774131	25118,4	8,581657	29006,3	9,824067	31035	11,894519
21	20932	7,711607	25118,4	8,556427	29006,3	9,87249	31035	11,747027
22	21030,5	7,812707	25236,6	8,38415	29006,3	9,712873	31035	11,328472
23	20932	8,047475	25118,4	8,809984	29006,3	9,798066	31035	11,311444
24	21030,5	7,881715	25118,4	8,538025	29006,3	9,702948	31035	11,520801
25	21030,5	7,724603	25177,4	8,632073	29141,4	9,656251	31035	11,711777
26	21030,5	7,54963	25118,4	8,599658	29006,3	9,45681	31035	11,725116
27	20932	7,514797	25280,4	8,7664	29006,3	9,729786	31035	11,316759
28	20932	7,811108	25177,4	8,454249	29006,3	9,551301	31035	11,487295
29	21030,5	7,659999	25118,4	8,666149	29006,3	9,225628	31035	11,719321
30	21052	7,672929	25118,4	8,494658	29140,1	10,044541	31035	11,302187
Médias	$\bar{\chi}$	$\bar{\tau}(s)$	$\bar{\chi}$	$\bar{\tau}(s)$	$\bar{\chi}$	$\bar{\tau}(s)$	$\bar{\chi}$	$\bar{\tau}(s)$
	20981,9	7,688756	25169,066667	8,547424	29034,653333	9,697821	31035	11,489392

Quadro 11 – Resultados da instância SF_21_38 da classe C1.

Execução	$\alpha = 0.5$		$\alpha = 0.6$		$\alpha = 0.7$		$\alpha = 0.8$	
	#min	$\tau(s)$	#min	$\tau(s)$	#min	$\tau(s)$	#min	$\tau(s)$
1	2439450	17,185216	2845420	23,237016	3123500	24,479752	3170900	21,809792
2	2439450	17,164579	2846900	22,911614	3123230	23,02137	3170900	21,577803
3	2440800	17,589506	2846900	21,817117	3123230	23,011568	3170900	21,390524
4	2440800	17,733865	2845420	21,826892	3123500	24,611541	3170900	21,723075
5	2440800	17,591015	2845420	22,719311	3123230	23,160815	3170900	21,731878
6	2440800	17,773068	2845420	21,830748	3123500	24,298814	3170900	21,099837
7	2440800	17,531004	2846900	22,893584	3123230	23,179939	3170900	21,784124
8	2444950	17,754019	2846900	22,442283	3123230	23,424774	3170900	22,517303
9	2439450	17,463617	2846900	23,478703	3123500	24,210809	3170900	22,001212
10	2439450	17,601131	2845420	23,370431	3123230	23,405883	3170900	21,879452
11	2440800	17,877791	2845420	21,402238	3123500	24,65421	3170900	21,612213
12	2440800	17,664341	2846900	23,141859	3123500	24,625723	3170900	22,222915
13	2439450	17,890658	2845420	22,781846	3123500	24,04317	3170900	21,258922
14	2439450	17,64692	2846900	21,56423	3123500	24,021545	3170900	21,644828
15	2439450	16,997239	2845420	22,275778	3123500	24,432057	3170900	22,22368
16	2440800	17,646676	2855980	19,311608	3123500	25,03638	3170900	21,883694
17	2440800	17,590635	2846900	21,636566	3123500	25,648888	3170900	21,549865
18	2439450	17,484137	2845420	22,314005	3123230	23,526184	3170900	22,111922
19	2440800	18,018459	2846900	22,708807	3123500	23,886606	3170900	22,179453
20	2449900	17,442212	2845420	22,607101	3123230	22,991778	3170900	21,952253
21	2464650	18,329774	2845420	22,301288	3123230	23,181681	3170900	20,979355
22	2440800	17,941063	2845420	20,853101	3123500	24,245606	3170900	21,60978
23	2440800	17,686767	2846900	22,375836	3123230	23,300681	3170900	21,559774
24	2440800	17,496238	2846900	22,429816	3123500	24,691573	3170900	21,57853
25	2463250	17,612741	2845420	22,214773	3123500	24,147275	3170900	21,259344
26	2439450	17,312609	2846900	21,645068	3123500	23,990121	3170900	20,722355
27	2439450	17,11075	2846900	21,275556	3123230	23,117198	3170900	21,335934
28	2439450	17,18921	2845420	21,011434	3123230	23,284036	3170900	21,23306
29	2440800	17,62162	2845420	21,243129	3123500	24,659458	3170900	21,160987
30	2439450	17,449404	2845420	21,991834	3123230	24,141962	3170900	21,577056
Médias	$\bar{\chi}$	$\bar{\tau}(s)$	$\bar{\chi}$	$\bar{\tau}(s)$	$\bar{\chi}$	$\bar{\tau}(s)$	$\bar{\chi}$	$\bar{\tau}(s)$
	2442245	17,579875	2846413,33	8,547424	22,120452	23,947713	3170900	21,639031

Quadro 12 – Resultados da instância MS_40_65 da classe C2.

Execução	$\alpha = 0.5$		$\alpha = 0.6$		$\alpha = 0.7$		$\alpha = 0.8$	
	#min	$\tau(s)$	#min	$\tau(s)$	#min	$\tau(s)$	#min	$\tau(s)$
1	143573	195,143536	165481,2	231,615525	181355,7	264,917015	188621	312,679073
2	143882	193,848428	165481,2	230,58875	181355,7	268,289153	188621	311,397866
3	143882	186,958871	165481,2	234,060129	181355,7	270,046464	188621	310,856441
4	143573	190,428374	165481,2	220,420954	181355,7	288,194331	188621	312,123849
5	143573	195,643071	165481,2	232,118051	181355,7	259,079002	188621	311,039769
6	143259	193,292886	165481,2	233,520709	181355,7	263,055372	188621	312,158952
7	143573	202,713519	165481,2	230,01954	181355,7	284,273705	188621	313,585694
8	143573	190,394427	165481,2	224,602801	181355,7	268,245035	188621	310,998544
9	143882	183,255129	165481,2	229,385536	181355,7	259,60627	188621	313,713754
10	143882	190,046302	165544,4	223,580685	181355,7	278,972968	188621	311,657234
11	143573	184,565329	165797,4	234,350848	181355,7	272,462314	188621	312,367015
12	143259	188,175298	165481,2	228,78902	181355,7	282,010741	188621	311,096613
13	143259	196,846596	165481,2	222,685227	181355,7	268,394632	188621	313,657613
14	143882	186,724233	165481,2	236,110832	181355,7	271,800972	188621	314,258613
15	143882	194,107353	165481,2	231,831574	181355,7	256,758426	188621	310,811881
16	143259	199,268881	165797,4	240,158366	181355,7	270,158388	188621	309,710845
17	143882	189,277657	165797,4	229,455768	181355,7	256,121137	188621	315,066037
18	143259	195,389239	165481,2	227,552433	181355,7	270,056619	188621	313,111716
19	143259	191,891357	165481,2	222,137365	181355,7	281,358536	188621	314,483822
20	145532,5	182,71632	165797,4	231,702455	181355,7	288,252333	188621	311,358868
21	143259	197,232039	165481,2	226,016373	181355,7	281,762825	188621	314,010585
22	143882	180,554666	165481,2	226,746835	181355,7	261,314176	188621	311,571778
23	143882	194,375159	165481,2	225,220665	181355,7	258,031452	188621	315,415631
24	143573	192,013072	165481,2	232,604333	181355,7	281,410431	188621	308,345116
25	143259	197,004614	165481,2	227,512632	181355,7	269,713606	188621	313,123812
26	143259	200,916301	165594,8	223,724085	181355,7	260,481147	188621	310,483458
27	143259	195,909334	165481,2	224,812135	181355,7	257,674502	188621	314,194795
28	143259	199,765394	165481,2	227,805693	181355,7	279,795553	188621	313,107818
29	143882	185,811894	165544,4	245,732459	181355,7	262,293637	188621	310,952829
30	143259	185,320008	165481,2	225,06037	181355,7	271,991526	188621	309,822987
Médias	$\bar{\chi}$	$\bar{\tau}(s)$	$\bar{\chi}$	$\bar{\tau}(s)$	$\bar{\chi}$	$\bar{\tau}(s)$	$\bar{\chi}$	$\bar{\tau}(s)$
	143615,716667	191,98631	165531,36	229,330738	181355,7	270,217409	188621	312,238767

Quadro 13 – Resultados da instância ME_40_99 da classe C2.

Execução	$\alpha = 0.5$		$\alpha = 0.6$		$\alpha = 0.7$		$\alpha = 0.8$	
	#min	$\tau(s)$	#min	$\tau(s)$	#min	$\tau(s)$	#min	$\tau(s)$
1	108114	481,261764	128047,8	527,598326	146431,3	569,381587	160488,8	694,528299
2	108046	475,69008	127949,6	525,35821	146431,3	595,458126	160218,2	695,051468
3	108711	452,951042	127949,6	485,662041	146943,6	578,08024	160596,4	686,565198
4	108352,5	469,843192	128047,8	531,86773	146884,9	616,82037	160233,8	684,795173
5	108426,5	472,53002	128409,6	529,577133	146647,2	563,525406	160191,8	697,107514
6	108210	464,287656	128038	547,333896	146431,3	590,239596	160233,8	645,644897
7	108168,5	459,846935	128047,8	528,694422	146431,3	546,775365	160191,8	673,494162
8	108231	452,275071	128035,4	509,428705	146647,2	587,242869	160506,2	703,831648
9	108168,5	456,039672	128498	488,242151	146661,9	636,219181	160191,8	681,301218
10	108352,5	480,505111	128409,6	528,476966	146661,9	632,677337	160191,8	691,475559
11	108114	486,613596	127983,4	514,857756	146431,3	616,930948	160506,2	706,088585
12	108155,5	466,983634	128244,8	509,748804	146431,3	577,648447	160191,8	716,365448
13	108114	466,833655	128033,2	531,768154	146431,3	557,962482	160449	698,882654
14	108175,5	477,089238	128113,4	532,242648	146833,2	554,311284	160511	690,878889
15	108168,5	478,299191	128646	518,059386	146647,2	604,336509	160449	689,349788
16	108266	465,873961	127949,6	524,287201	147151,4	586,843613	160233,8	626,838429
17	108231	461,701292	128047,8	530,171412	147068,7	593,368249	160191,8	698,237897
18	108046	481,754101	128229,6	534,225786	146824,1	600,207561	160191,8	695,235693
19	108168,5	458,292188	127949,6	532,353852	146431,3	581,700154	160191,8	704,54452
20	108104,5	481,195184	128113,4	517,73902	146884,9	633,84354	160489	703,221323
21	108255,5	464,639753	128408	531,370556	146431,3	591,250593	160506,2	710,024876
22	108266	452,719352	128245,6	491,5992	146431,3	554,454293	160449	703,40553
23	108255,5	462,45714	128113,4	544,415015	146431,3	604,473294	160191,8	673,596351
24	108168,5	474,493067	128035,4	531,693831	146824,1	616,438876	160488,8	675,487483
25	108168,5	477,939819	128026	526,718796	146431,3	550,266715	160191,8	676,690364
26	108195,5	462,767757	127983,4	528,748428	146824,1	576,881028	160233,8	699,729816
27	108147	490,380151	128245,6	519,521409	146431,3	605,589304	160191,8	673,326562
28	108714,5	483,928575	128356,4	535,340649	146647,2	589,56118	160191,8	688,435099
29	108583	487,936102	127949,6	520,59225	146431,3	580,480204	160191,8	674,981766
30	108265,5	470,147304	128038	562,473191	146431,3	589,638421	160346	693,565219
Médias	\bar{X}	$\bar{\tau}(s)$	\bar{X}	$\bar{\tau}(s)$	\bar{X}	$\bar{\tau}(s)$	\bar{X}	$\bar{\tau}(s)$
	108244,783333	470,575853	128138,18	524,672231	146620,703333	589,420226	160314,413333	688,422714

Quadro 14 – Resultados da instância MD_40_164 da classe C2.

Execução	$\alpha = 0.5$		$\alpha = 0.6$		$\alpha = 0.7$		$\alpha = 0.8$	
	#min	$\tau(s)$	#min	$\tau(s)$	#min	$\tau(s)$	#min	$\tau(s)$
1	92269,5	1272,774047	110727	1343,843233	129320,2	1530,479024	145084,2	2068,609908
2	92350,5	1322,72203	110863,2	1431,808219	129359,7	1626,246028	144764,4	2189,99433
3	92388,5	1258,591893	110748,6	1416,639292	128945,6	1546,508582	144772,4	2221,745061
4	92274,5	1331,15203	110731,8	1312,125629	129129,4	1573,850495	144759	2041,264363
5	92214,5	1261,548352	110888,4	1377,314567	129228,5	1595,99001	144819,4	2053,12829
6	92320,5	1359,035409	110793,6	1448,447767	129075,6	1468,078678	144730,8	1972,752044
7	92498,5	1293,24816	110696,4	1412,345996	129160,5	1606,902532	144743	2090,825054
8	92284	1353,833571	110524,8	1462,325373	128960,7	1520,005231	144743	2117,469782
9	92224	1309,039358	110769	1368,75304	129081,4	1612,356096	144991,2	2155,932675
10	92292	1318,280482	110917,8	1435,52909	129189,3	1686,624799	145051,4	2327,743521
11	92348	1346,837555	110653,8	1313,798694	129001,1	1691,017656	144922	2085,842718
12	92104	1291,902632	110901,6	1417,471876	129202,1	1678,822683	144773,4	2122,323391
13	92173	1353,192931	110569,2	1372,292383	129122,4	1725,561817	144743	2136,173969
14	92247	1330,757676	110740,8	1410,194443	129046,8	1628,604984	144718,2	2200,439876
15	92486,5	1364,576277	110947,8	1483,693752	129163	1661,241688	144676	2211,489282
16	92302	1348,201749	110887,2	1495,125113	129312,4	1674,802978	144743	2151,113704
17	92314,5	1323,877699	110641,2	1434,183231	129249,7	1886,833087	144676	2014,967214
18	92244	1426,089741	110700	1413,397464	129330,9	1672,225018	144772,4	2075,719257
19	92291,5	1421,171995	110728,8	1387,412324	129126,7	1689,243866	145059,2	1951,489721
20	92287,5	1348,293965	110718,6	1449,213886	129291,3	1797,860452	144773,4	2148,037479
21	92207,5	1375,693089	110639,4	1369,316133	129111,3	1696,682688	144743	2020,924602
22	92217,5	1277,063329	110770,8	1541,218883	129175,9	1768,913275	144759	2118,896459
23	92498	1308,817668	110556,6	1530,896959	129159,1	1824,880262	144922	1901,620512
24	92235	1352,539025	110750,4	1499,059681	129181	1775,335014	144789,6	2127,633539
25	92426,5	1305,384682	110556,6	1338,789956	129271	1669,521101	144759	2128,109
26	92364	1297,260065	110715,6	1379,860141	128935	1548,74905	144676	1859,119311
27	92280	1337,642146	110689,2	1413,08531	129213,3	1960,688451	144676	2067,34033
28	92274	1347,122088	110740,2	1467,448081	129165	1634,320155	144789,6	2033,098572
29	92468,5	1329,565476	110854,2	1422,607908	129096,7	1702,774588	144743	2068,016315
30	92358,5	1338,314313	110722,2	1436,978648	129177,3	1777,6964	144676	2033,686783
Médias	\bar{X}	$\tau(s)$	\bar{X}	$\tau(s)$	\bar{X}	$\tau(s)$	\bar{X}	$\tau(s)$
	92308,133333	1330,150981	110738,16	1419,505902	129159,43	1674,427223	144794,953333	2089,850235

APÊNDICE B – Grafos que Representam as Instâncias

Neste Apêndice são mostrados os grafos que representam as instâncias, e também, os grafos resultantes de como ficou a configuração da malha viária das melhores soluções encontradas pelo VNS para cada experimento. Na representação de um grafo, se tem um arco não direcionado, quer dizer que esse arco representa um via de mão dupla da malha viária, e se o arco é direcionado, quer dizer que esse arco está representando uma via de mão única.

Figura 15 – Grafo da Solução inicial para a instância PQ_14_20.

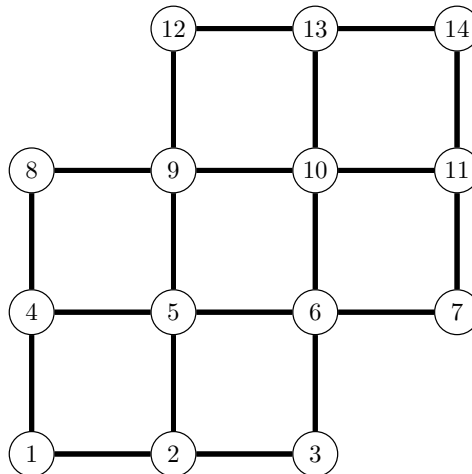


Figura 16 – Grafo da melhor solução encontrada para a instância PQ_14_20 com $\alpha = 0.8$.

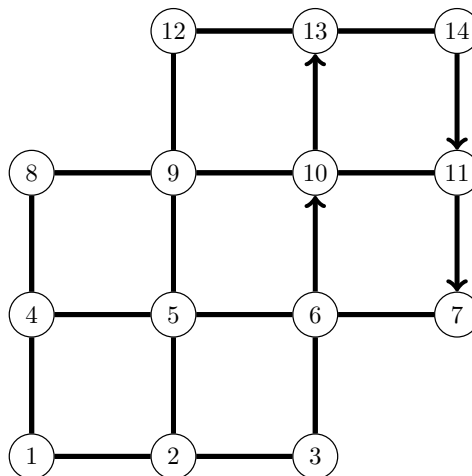


Figura 17 – Grafo da melhor solução encontrada para a instância PQ_14_20 com $\alpha = 0.7$.

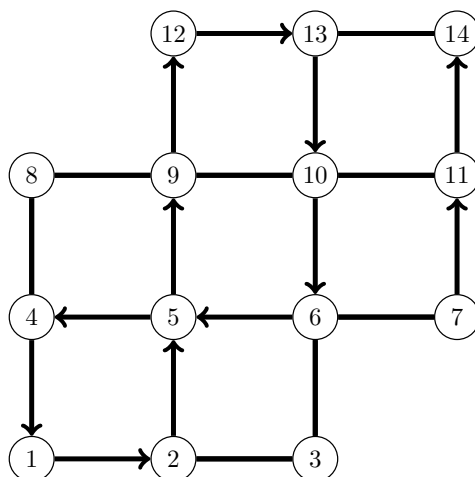


Figura 18 – Grafo da melhor solução encontrada para a instância PQ_14_20 com $\alpha = 0.6$.

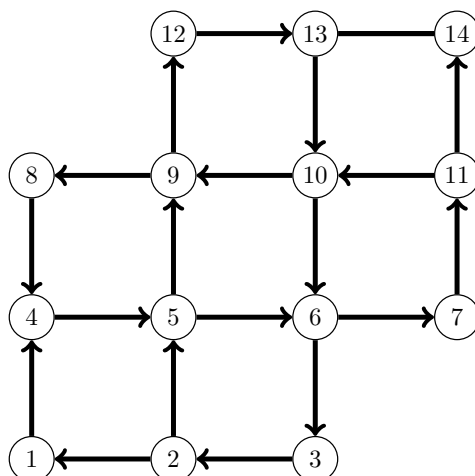


Figura 19 – Grafo da melhor solução encontrada para a instância PQ_14_20 com $\alpha = 0.5$.

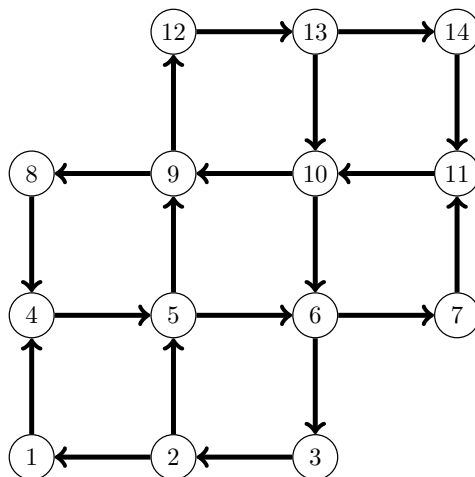


Figura 20 – Grafo da Solução inicial para a instância PQ_18_41.

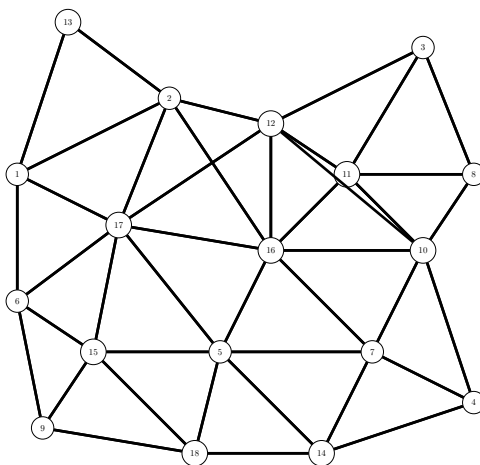


Figura 21 – Grafo da melhor solução encontrada para a instância PQ_18_41 com $\alpha = 0.8$.

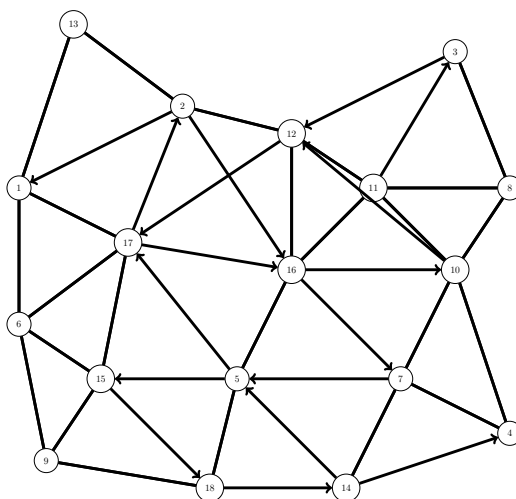


Figura 22 – Grafo da melhor solução encontrada para a instância PQ_18_41 com $\alpha = 0.7$.

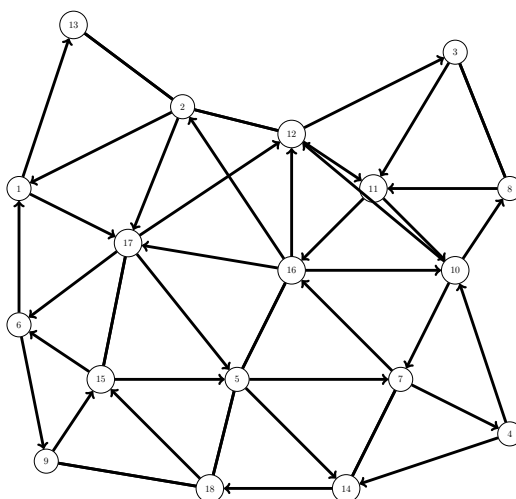


Figura 23 – Grafo da melhor solução encontrada para a instância PQ_18_41 com $\alpha = 0.6$.

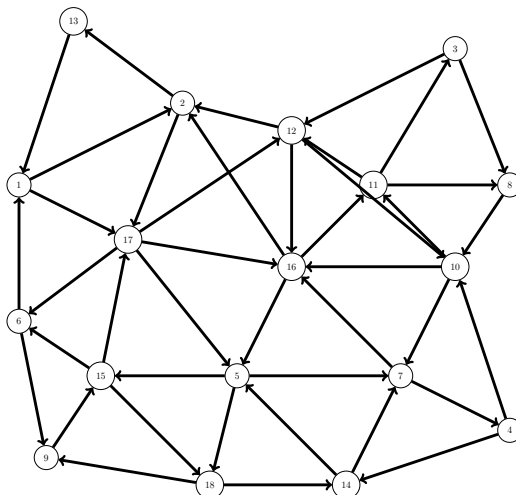


Figura 24 – Grafo da melhor solução encontrada para a instância PQ_18_41 com $\alpha = 0.5$.

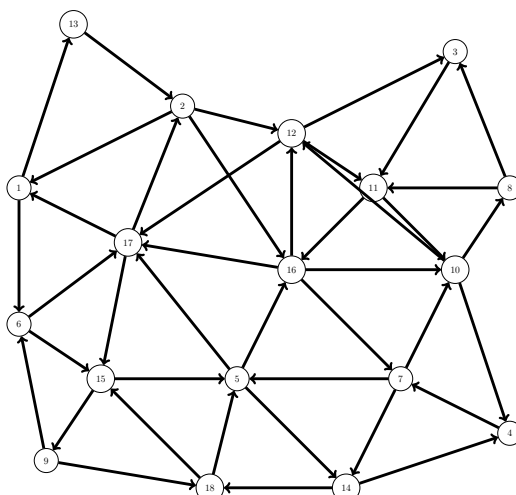


Figura 25 – Grafo da Solução inicial para a instância SF_21_38.

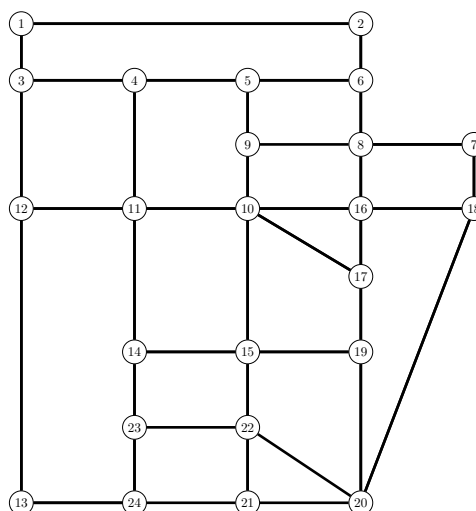


Figura 26 – Grafo da melhor solução encontrada para a instância SF_21_38 com $\alpha = 0.8$.

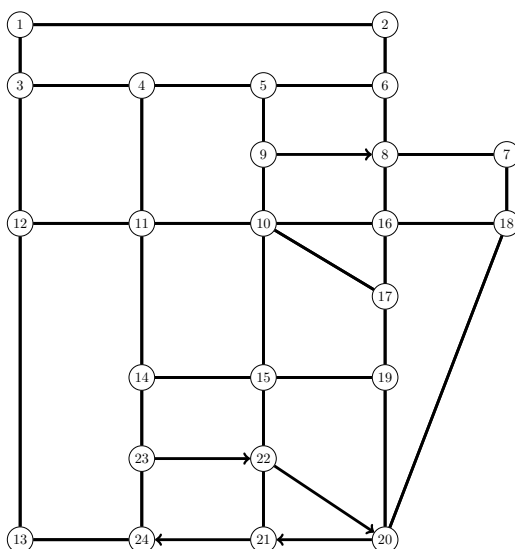


Figura 27 – Grafo da melhor solução encontrada para a instância SF_21_38 com $\alpha = 0.7$.

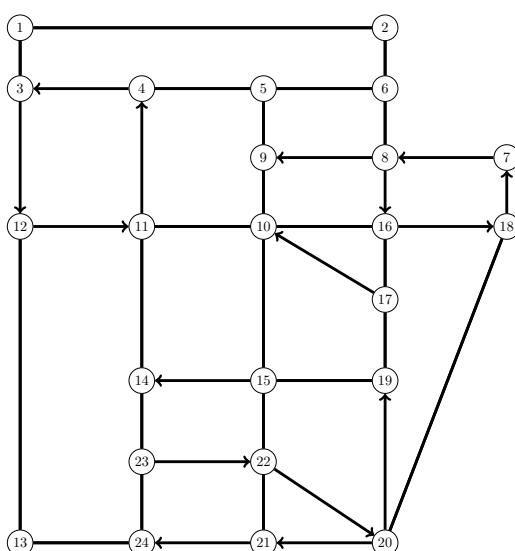


Figura 28 – Grafo da melhor solução encontrada para a instância SF_21_38 com $\alpha = 0.6$.

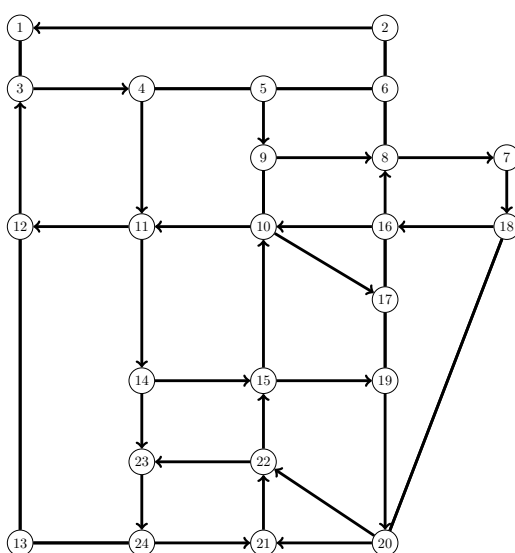


Figura 29 – Grafo da melhor solução encontrada para a instância SF_21_38 com $\alpha = 0.5$.

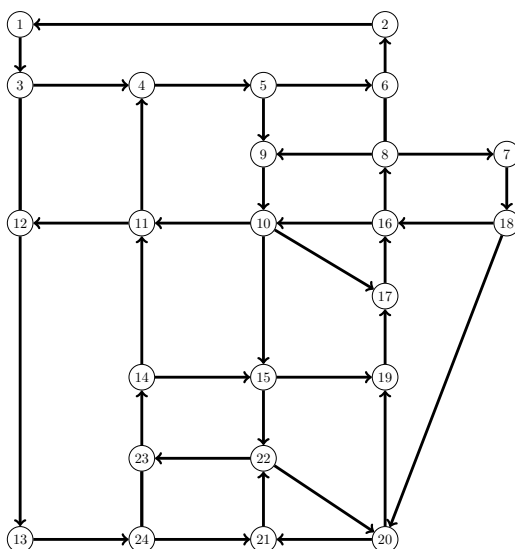


Figura 30 – Grafo da Solução inicial para a instância MS_40_65.

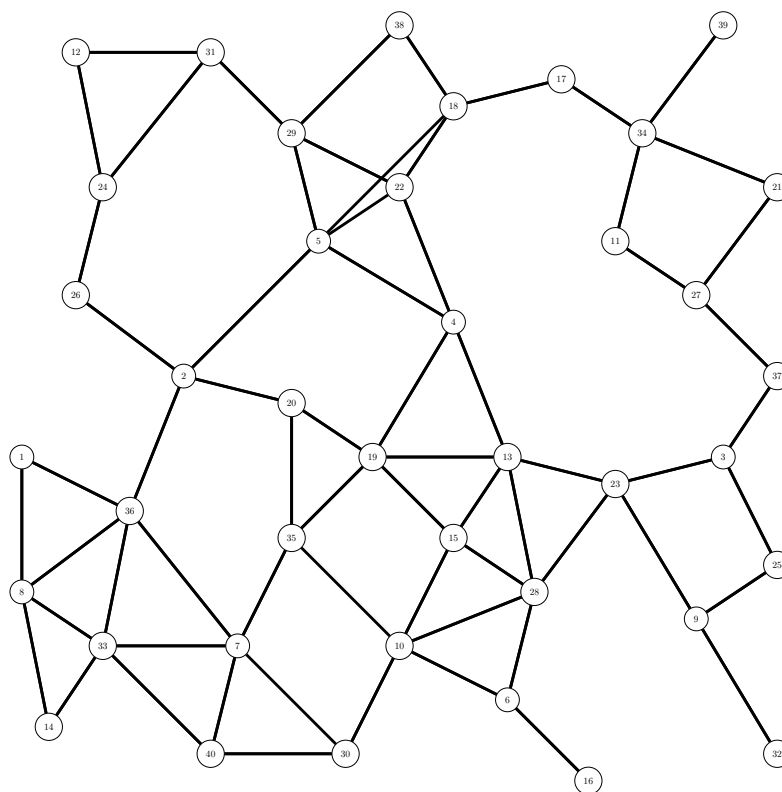


Figura 31 – Grafo da melhor solução encontrada para a instância MS_40_65 com $\alpha = 0.8$.

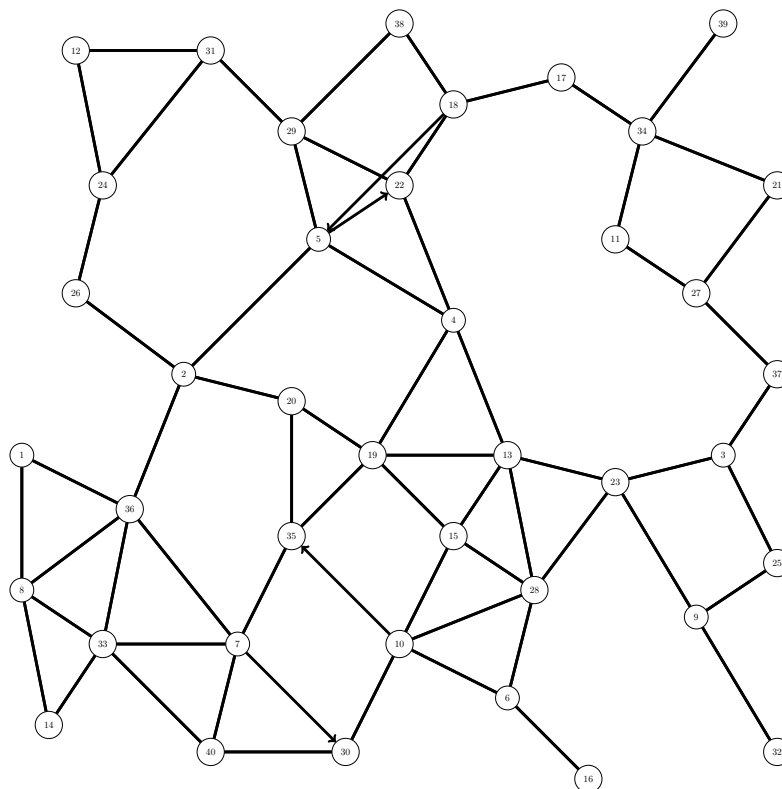


Figura 32 – Grafo da melhor solução encontrada para a instância MS_40_65 com $\alpha = 0.7$.

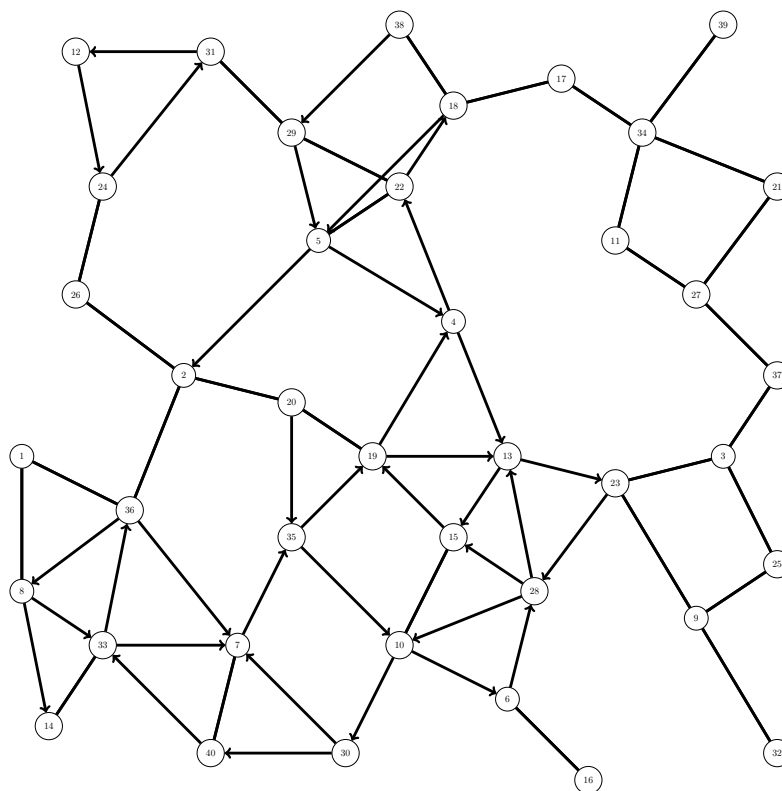


Figura 33 – Grafo da melhor solução encontrada para a instância MS_40_65 com $\alpha = 0.6$.

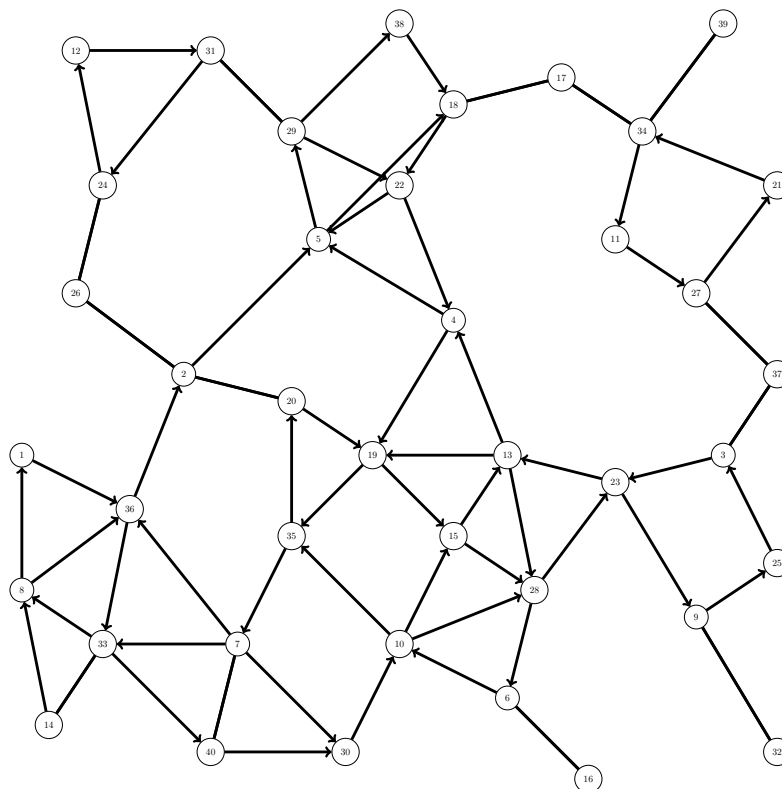


Figura 34 – Grafo da melhor solução encontrada para a instância MS_40_65 com $\alpha = 0.5$.

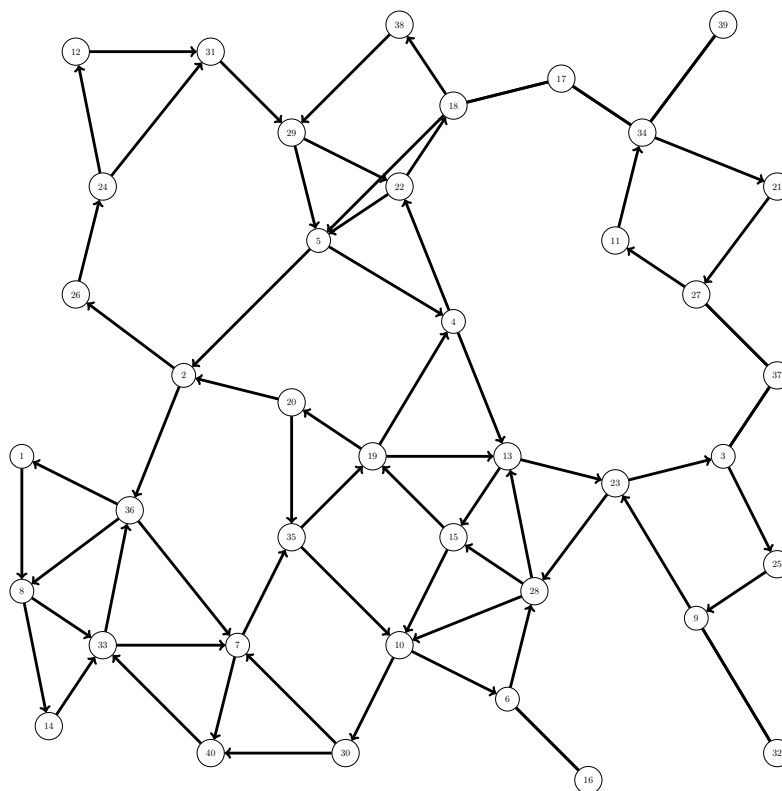


Figura 35 – Grafo da Solução inicial para a instância ME_40_99.

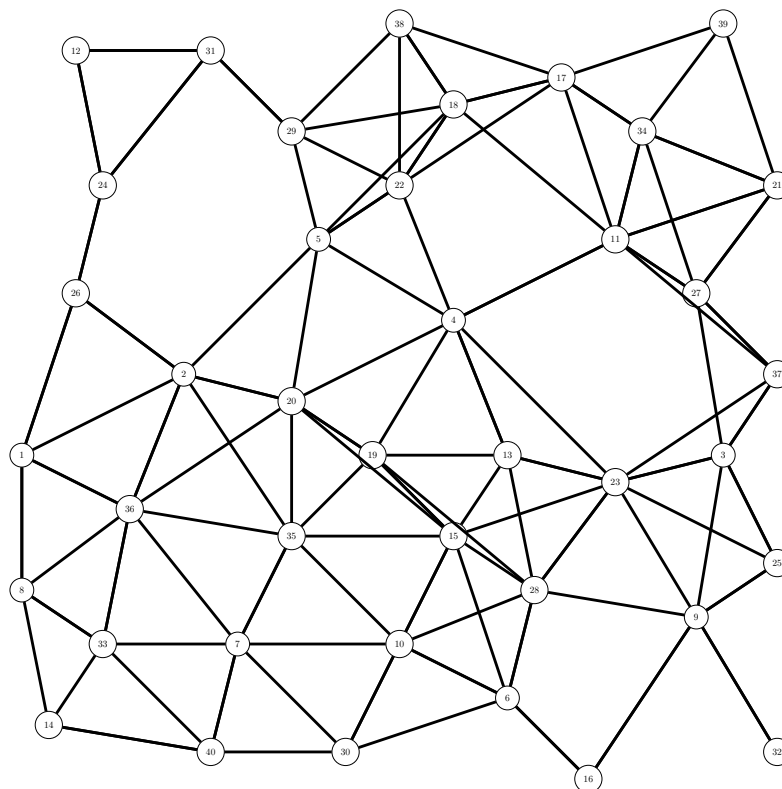


Figura 36 – Grafo da melhor solução encontrada para a instância ME_40_99 com $\alpha = 0.8$.

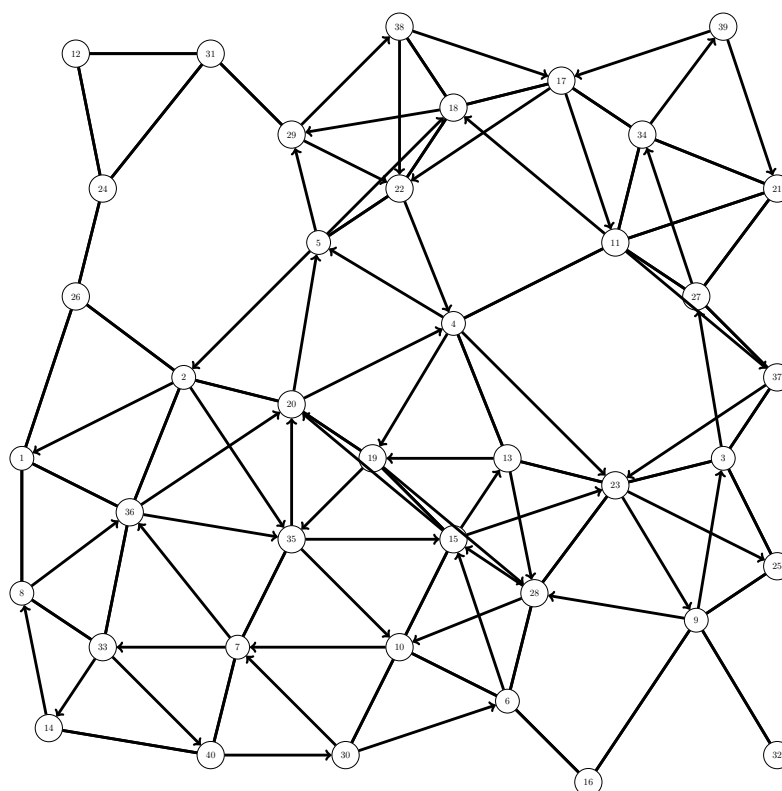


Figura 37 – Grafo da melhor solução encontrada para a instância ME_40_99 com $\alpha = 0.7$.

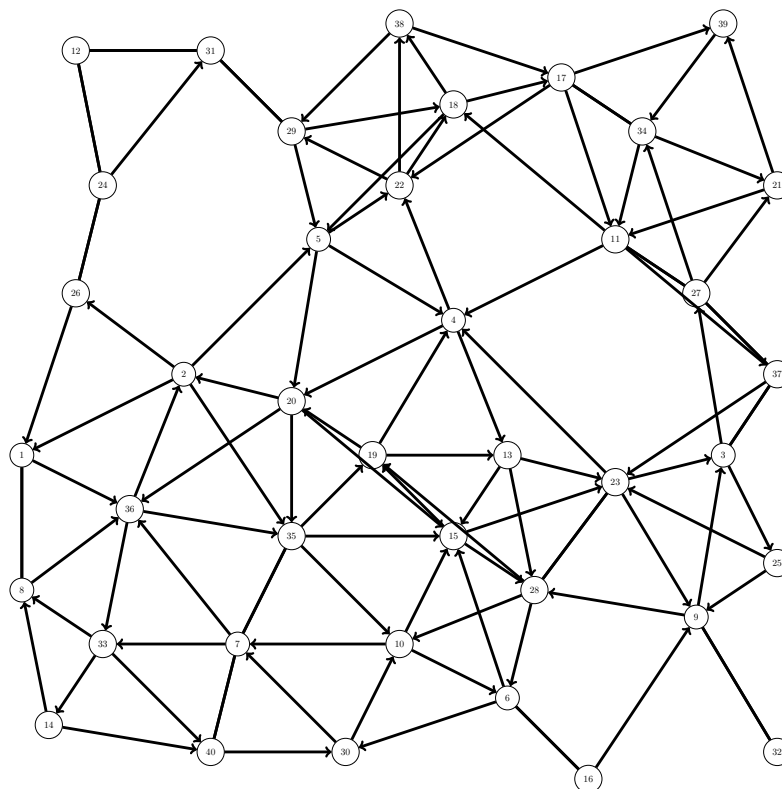


Figura 38 – Grafo da melhor solução encontrada para a instância ME_40_99 com $\alpha = 0.6$.

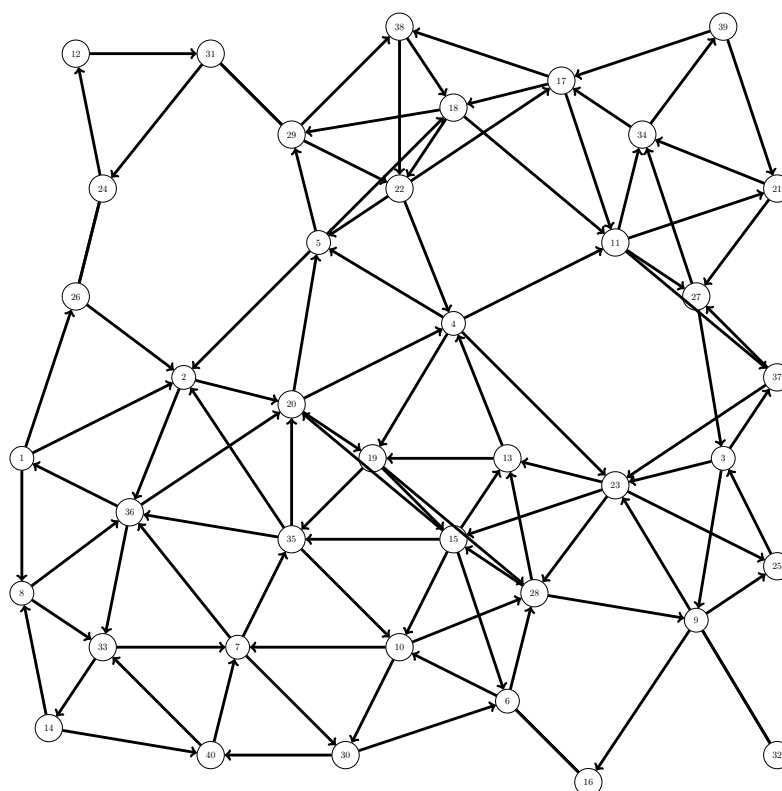


Figura 39 – Grafo da melhor solução encontrada para a instância ME_40_99 com $\alpha = 0.5$.

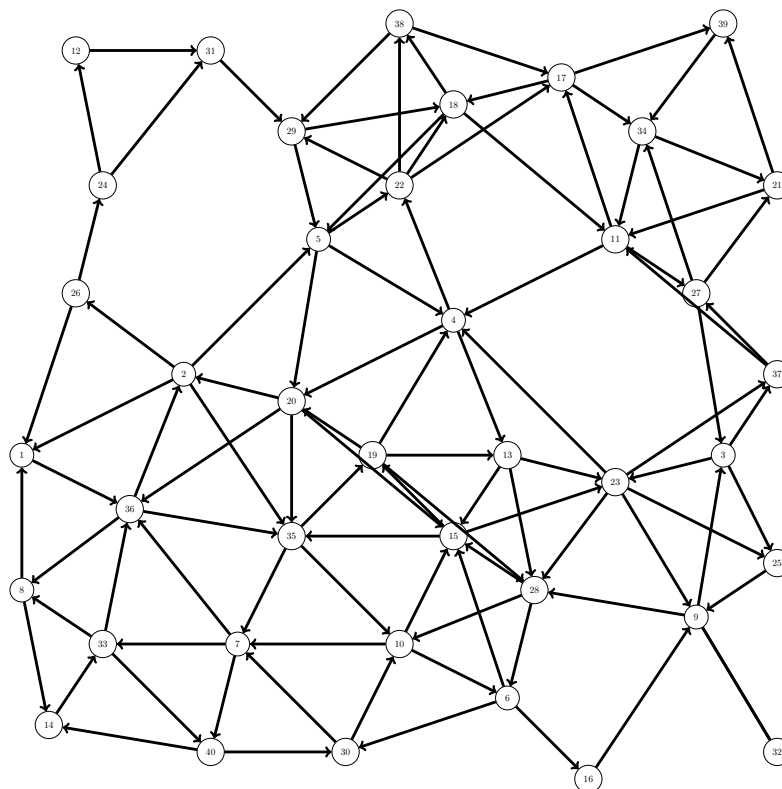


Figura 40 – Grafo da Solução inicial para a instância MD_40_164.

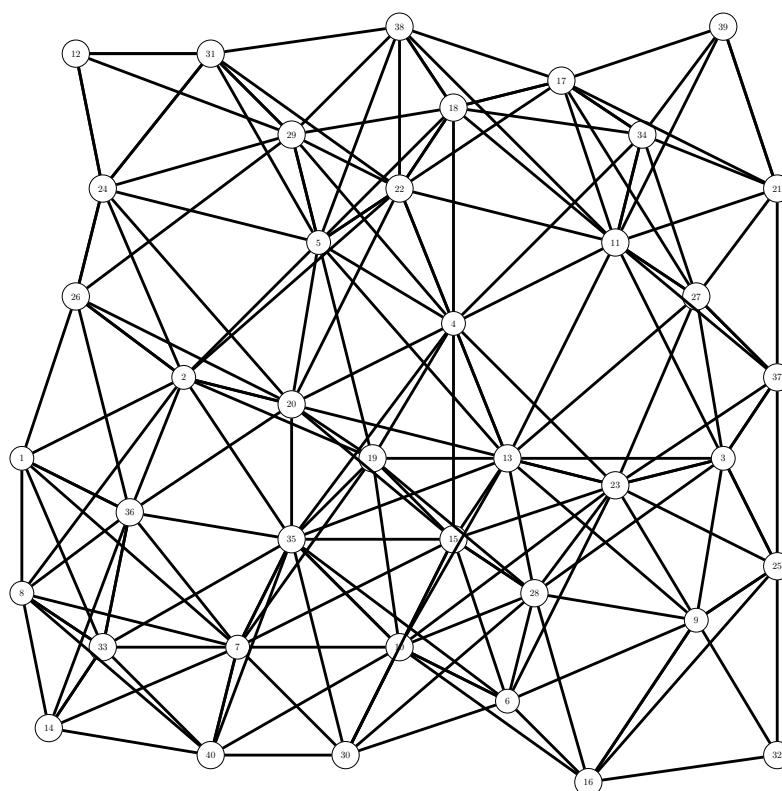


Figura 41 – Grafo da melhor solução encontrada para a instância MD_40_164 com $\alpha = 0.8$.

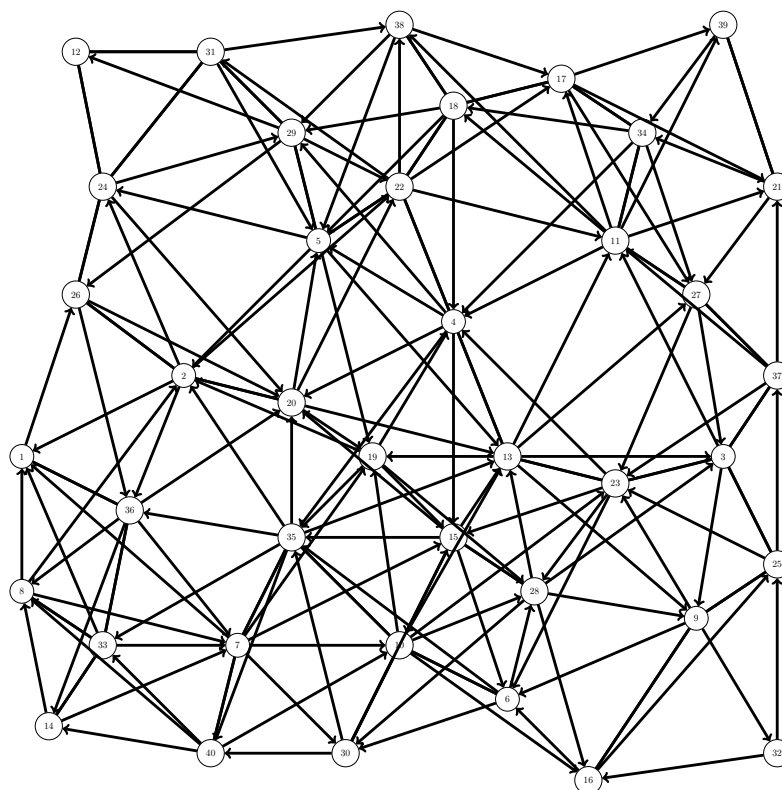


Figura 42 – Grafo da melhor solução encontrada para a instância MD_40_164 com $\alpha = 0.7$.

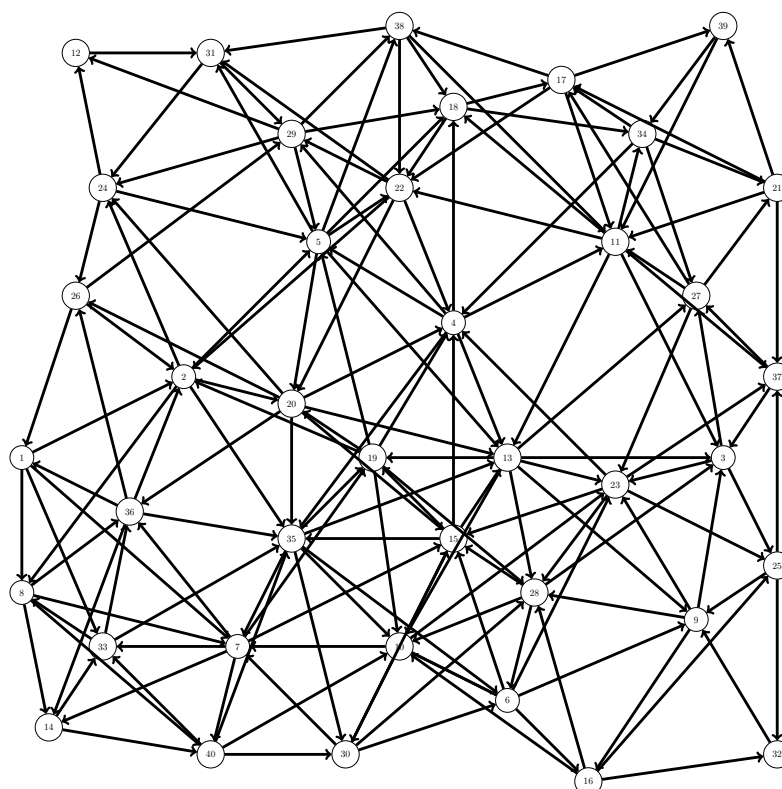


Figura 43 – Grafo da melhor solução encontrada para a instância MD_40_164 com $\alpha = 0.6$.

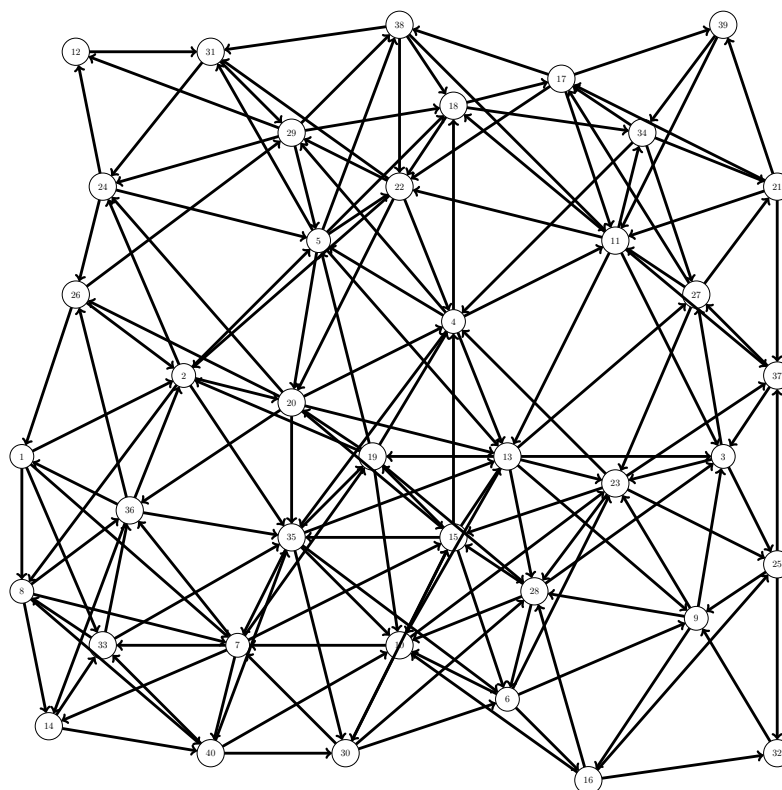


Figura 44 – Grafo da melhor solução encontrada para a instância MD_40_164 com $\alpha = 0.5$.

