

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ  
DEPARTAMENTO ACADÊMICO DE CONSTRUÇÃO CIVIL  
CURSO DE ENGENHARIA DE PRODUÇÃO CIVIL**

**WESLEY JOSÉ NOGUEIRA MEDEIROS**

**ABORDAGEM HEURÍSTICA PARALELIZADA PARA A RESOLUÇÃO  
DO PROBLEMA DO ROTEAMENTO DE VEÍCULOS CAPACITADOS  
COM BASE NA ESTRATÉGIA “*CLUSTER FIRST ROUTE SECOND*”**

**TRABALHO DE CONCLUSÃO DE CURSO**

**CURITIBA**

**2011**

**WESLEY JOSÉ NOGUEIRA MEDEIROS**

**ABORDAGEM HEURÍSTICA PARALELIZADA PARA A RESOLUÇÃO  
DO PROBLEMA DO ROTEAMENTO DE VEÍCULOS CAPACITADOS  
COM BASE NA ESTRATÉGIA “*CLUSTER FIRST ROUTE SECOND*”**

Trabalho de Conclusão de Curso de graduação, apresentado à Disciplina Projeto Final 2, do Curso Superior de Engenharia de Produção Civil do Departamento Acadêmico de Construção Civil – DACOC – da Universidade Tecnológica Federal do Paraná - UTFPR, como requisito parcial para obtenção do título de Engenheiro.

Orientador: Prof. M.Sc. Thiago André Guimarães

**CURITIBA**

**2011**

---

Dados Internacionais de Catalogação na Publicação

---

M488 Medeiros, Wesley José Nogueira  
Abordagem heurística paralelizada para a resolução do problema do roteamento de veículos capacitados com base na estratégia “*cluster first route second*” / Wesley José Nogueira Medeiros. — 2011.  
60 p. : il. ; 30 cm

Orientador: Thiago André Guimarães.

Trabalho de conclusão de Curso (Graduação) – Universidade Tecnológica Federal do Paraná. Curso Superior de Engenharia da Produção Civil, Curitiba, 2011.

Bibliografia: p. 44-46.

1. Problema do roteamento de veículo. 2. Computadores paralelos. 3. Heurística. I. Guimarães, Thiago André, orient. II. Universidade Tecnológica Federal do Paraná. Curso Superior de Engenharia da Produção Civil. III. Título.

CDD (22. ed.) 670.42

---

---

## FOLHA DE APROVAÇÃO

### **ABORDAGEM HEURÍSTICA PARALELIZADA PARA A RESOLUÇÃO DO PROBLEMA DO ROTEAMENTO DE VEÍCULOS CAPACITADOS COM BASE NA ESTRATÉGIA “CLUSTER FIRST ROUTE SECOND”**

Por

**WESLEY JOSÉ NOGUEIRA MEDEIROS**

Trabalho de Conclusão de Curso apresentado ao Curso de Engenharia de Produção Civil, da Universidade Tecnológica Federal do Paraná, defendido e aprovado em 23 de novembro de 2011, pela seguinte banca de avaliação:

---

Prof. Orientador – Thiago André Guimarães, MSc.  
UTFPR

---

Prof. Maiko Fernandes Buzzi, MSc.  
FAE

---

Prof. Arildo Dirceu Cordeiro, Dr.  
UTFPR

---

Prof. César Augusto Romano, Dr.  
UTFPR

À memória de minha mãe, Irene Martins Nogueira Medeiros, que tão sabiamente direcionou meus estudos ao valorizar a disciplina, a dedicação e a persistência como instrumentos fundamentais para o sucesso profissional.

## **AGRADECIMENTOS**

- Ao meu pai, José Medeiros, que trabalhou incansavelmente em meio às adversidades para garantir ao seu filho uma boa infraestrutura familiar e uma educação de qualidade, sem os quais eu nunca teria alcançado o tão desejado título de Engenheiro.
- Ao meu orientador, o professor Thiago André Guimarães, pela orientação, paciência e principalmente pelo otimismo nas palavras de incentivo e reconhecimento que tanto me encorajaram na condução deste trabalho.
- A minha namorada Talita Pouzo Minatel, pela paciência e companheirismo dispensados durante a minha trajetória educacional na UTFPR.
- Ao meu amigo Luis Gustavo Pereira, pelas valiosas aulas de programação em C#, LINQ e PLINQ sem as quais a implementação computacional desse trabalho não seria possível.
- Aos profissionais da Sala de Informações Aeronáuticas do Aeroporto Internacional Afonso Pena (DTCEA-CT), meus grandes amigos, pela imensa cooperação ao longo dos meus anos de estudo na UTFPR.

*“Inovação e progresso são alcançados somente por aqueles que se aventuram além do procedimento operacional padrão.”*

(D. Michael Abrashoff – 2006)

*“A responsabilidade permanente de todo Gerente de Produção é melhorar o desempenho de suas operações.”*

(Nigel Slack – 2009)

## RESUMO

MEDEIROS, W. J. N. **Abordagem Heurística Paralelizada para a Resolução do Problema do Roteamento de Veículos Capacitados com base na Estratégia “Cluster First Route Second”**. 2011. 60p. Trabalho de Conclusão de Curso, Universidade Tecnológica Federal do Paraná, UTFPR, Curitiba.

O Problema de Roteamento de Veículos Capacitados (PRVC) busca determinar um conjunto de rotas para uma frota homogênea de veículos, partindo de um depósito central com destino a um conjunto de clientes que demandam determinado produto. Por ser um dos mais importantes e complexos problemas afetos à logística de distribuição, alternativas heurísticas para o problema são constantemente apresentadas. Nesse sentido, este trabalho apresenta uma abordagem heurística baseada na estratégia “*Cluster First Route Second*” que consiste primeiramente em agrupar os clientes conforme a demanda para posteriormente construir rotas para os grupos formados. A abordagem emprega os algoritmos clássicos de Teitz e Bart (1968) e Gillett e Johnson (1976) para a resolução do problema do agrupamento, enquanto que as rotas para cada grupo são construídas pela heurística do vizinho mais próximo e refinadas através da heurística de melhoria 2-opt. Foram empregadas técnicas de processamento paralelo na estruturação do software desenvolvido, objetivando à redução do tempo de cálculo computacional. Testes realizados com as instâncias clássicas disponíveis na literatura apontaram um desvio médio de 15,0% em relação aos resultados ótimos dos problemas, com a vantagem de consumir um tempo de processamento computacional inferior a 7 segundos para a resolução de cada instância.

**Palavras-Chave:** Roteamento de Veículos Capacitados; Métodos Heurísticos; Computação Paralela.



## **ABSTRACT**

MEDEIROS, W. J. N. **Parallel Heuristic Approach to solve the Capacitated Vehicle Routing Problem based on cluster first route second strategy.** 2011. 60p. Federal Technology University of Paraná, UTFPR, Brazil.

The Capacitated Vehicle Routing Problem (CVRP) seeks to determine a set of routes for a homogeneous fleet of vehicles from a central warehouse bound to a set of customers who demands products. As one of the most important and complex problems related to the logistics of distribution, heuristics to the problem are constantly developed. On this way, this paper addresses the heuristic approach based on strategy "Cluster First Route Second" that consists of grouping customers according to demand to further build routes for the groups formed. The approach employs classical algorithms Teitz and Bart (1968) and Gillett and Johnson (1976) for solving the clustering problem, while the routes for each group are constructed by the nearest neighbor heuristic and refinements arising from exchanges by 2-opt. It is also used parallel computing to reduce the computational time. Simulations performed with the classic instances available in the shown literature results on a performance of 15,0% instead. The results took an advantage of consuming a computational processing time shorter than 7 seconds to solve each instance.

**Key-Words:** Capacited Vehicle Routing Problem, Heuristics Methods, Parallel Computing

## LISTA DE FIGURAS

Figura 1 - Utilização da classe <i>Parallel</i> em trecho de código paralelizado.....	19
Figura 2 - Exemplo de aplicação de PLINQ no <i>software</i> criado. ....	20
Figura 3 - Instância exemplo do PRVC. ....	23
Figura 4 - Heurística de Teitz e Bart.....	26
Figura 5 - Algoritmo de Gillett e Johnson para agrupamento. ....	27
Figura 6 - Algoritmo do Vizinho Mais Próximo (VMP). ....	28
Figura 7 - Troca 2-opt entre os trechos 1-2 e 3-4.....	29
Figura 8 - Heurística proposta para a resolução do PRVC .....	30
Figura 9 - Pseudocódigo para a heurística proposta.....	30
Figura 10 - Interface do usuário para o <i>software</i> desenvolvido.....	33
Figura 11 - Exemplo do gráfico evolutivo Custo x Iteração .....	34
Figura 12 - Taxa de ocupação dos núcleos de processamento computacional .....	35
Figura 13 - Exemplo de tabela com as rotas dos veículos .....	36
Figura 14 - Exemplo de solução gráfica 2D para as rotas dos veículos.....	36
Figura 15 - Solução gráfica para a instância M-n101-k10 .....	41
Figura 16 - Solução gráfica para a instância B-n31-k5.....	41
Figura 17 - Solução gráfica para a instância E-n30-k3.....	42

## LISTA DE SIGLAS E ACRÔNIMOS

LINQ	<i>Language Integrated Query</i>
OVRP	<i>Open Vehicle Routing Problem</i>
Obs.	Observação
PCV	Problema do Caixeiro Viajante
PLINQ	<i>Parallel Language Integrated Query</i>
PRV	Problema de Roteamento de Veículos
PRVC	Problema de Roteamento de Veículos Capacitados
UTFPR	Universidade Tecnológica Federal do Paraná
VMP	Vizinho mais Próximo
UI	<i>User Interface</i>

## LISTA DE TABELAS

Tabela 1 - Instâncias utilizadas para avaliação da heurística proposta.....	31
Tabela 2 - Resultados para as instâncias da classe A.....	37
Tabela 3 - Resultados para as instâncias da classe B.....	38
Tabela 4 - Resultados para as instâncias da classe E.....	38
Tabela 5 - Resultado para a instância da classe G.....	39
Tabela 6 - Resultados para as instâncias da classe M.....	39
Tabela 7 - Resultados para as instâncias da classe P.....	39
Tabela 8 - Resultados para a instância da classe V.....	40
Tabela 9 - Desvios médios obtidos na resolução do PRVC.....	40

# SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b> .....	<b>14</b>
1.1	JUSTIFICATIVA.....	14
1.2	OBJETIVOS.....	15
1.3	METODOLOGIA .....	15
1.4	ESTRUTURA .....	16
<b>2</b>	<b>REFERENCIAL TEÓRICO</b> .....	<b>17</b>
2.1	O PROBLEMA DO ROTEAMENTO DE VEÍCULOS CAPACITADOS (PRVC).....	17
2.2	A ESTRATÉGIA “CLUSTER FIRST ROUTE SECOND” .....	18
2.3	PROCESSAMENTO COMPUTACIONAL PARALELO .....	19
2.4	ESTRATÉGIAS PARA RESOLUÇÃO DO PRVC .....	21
<b>3</b>	<b>MATERIAIS E MÉTODOS</b> .....	<b>23</b>
3.1	DESCRIÇÃO FORMAL DO PRVC .....	23
3.2	ESTRATÉGIA DE RESOLUÇÃO.....	25
3.2.1	O problema das $p$ -medianas.....	25
3.2.2	O problema da formação de grupos de clientes (clusterização) .....	27
3.2.3	O problema da roteirização: construção de uma solução Inicial.....	27
3.2.4	O problema da roteirização: refinamento da solução inicial.....	28
3.3	HEURÍSTICA PROPOSTA PARA A RESOLUÇÃO DO PRVC.....	29
<b>4</b>	<b>RESULTADOS E DISCUSSÕES</b> .....	<b>33</b>
4.1	O SOFTWARE DESENVOLVIDO.....	33
4.2	RESULTADOS OBTIDOS PARA AS INSTÂNCIAS.....	36
<b>5</b>	<b>CONCLUSÕES</b> .....	<b>43</b>
	<b>REFERÊNCIAS</b> .....	<b>44</b>
	<b>APÊNDICES</b> .....	<b>47</b>

# 1 INTRODUÇÃO

As técnicas de pesquisa operacional são largamente utilizadas para a gestão eficiente das operações empresariais. No segmento de Logística e *Supply Chain Management* elas são fundamentais para a distribuição dos produtos aos clientes uma vez que auxiliam a determinação das melhores rotas para os veículos.

Na área de determinação de rotas, um dos problemas frequentemente estudados em pesquisa operacional é o Problema de Roteamento de Veículos Capacitados (PRVC). Por associar a capacidade de cada veículo como um fator restritivo, este problema surge como uma importante abordagem a ser empregada para melhorar a distribuição de produtos e minimizar os custos de transportes das empresas.

Ao pesquisar esse problema, o presente trabalho apresenta uma alternativa de resolução do PRVC pautada em técnicas de pesquisa operacional atuais, no intuito de contribuir com as discussões sobre o tema e desenvolver ainda mais as ferramentas tecnológicas empregadas na área de Logística e *Supply Chain Management*.

## 1.1 JUSTIFICATIVA

Chopra e Meindl (2010) cita que o sucesso da cadeia de suprimentos em uma empresa está intrinsecamente associado ao uso eficiente de transporte. O mesmo autor apresenta o transporte como um componente significativo dos custos incorrido por grande parte das cadeias de abastecimento.

No mesmo sentido, Slack *et al.* (2009) ressalta que independente do modal de transporte a ser utilizado, a designação das rotas a serem seguidas pelos veículos é uma das decisões mais difíceis a serem tomadas por quem administra uma rede de distribuição. Tal dificuldade justifica-se pelo impacto gerado pela decisão no respectivo nível de reposta da empresa às demandas dos clientes.

Dessa forma, torna-se pertinente o desenvolvimento de alternativas computacionais que auxiliem os gestores na resolução dos problemas de designação e roteamento de veículos.

Para tanto, o presente trabalho justifica-se pela iniciativa de incrementar o rol das heurísticas atualmente aplicadas na resolução do PRVC e buscar o desenvolvimento de novas tecnologias que consumam tempos de processamentos computacionais menores.

Esta pesquisa justifica-se ainda pela importância de propor e avaliar, à luz das técnicas de Pesquisa Operacional, uma nova abordagem heurística para a resolução de um importante problema de distribuição em redes logísticas.

## 1.2 OBJETIVOS

O objetivo deste trabalho é propor e avaliar uma nova abordagem heurística para a resolução do Problema de Roteamento de Veículos Capacitados (PRVC). Esta abordagem será baseada na estratégia “*Cluster First Route Second*” e empregará heurísticas clássicas e técnicas de paralelismo computacional para a redução do tempo de processamento necessário à resolução do PRVC.

A fim de alcançar o objetivo geral, propõem-se os seguintes objetivos específicos:

- a) Identificar os principais conceitos relacionados ao Problema do Roteamento de Veículos Capacitados e as técnicas atualmente utilizadas para a resolução do mesmo;
- b) Apresentar uma heurística para a resolução do PRVC e implementá-la computacionalmente, em plataforma *open source*, visando subsidiar uma análise adequada da estratégia proposta;
- c) Analisar os resultados obtidos através da implementação computacional da heurística no sentido de validar a estratégia empregada;
- d) Verificar os impactos da abordagem proposta no tempo de processamento computacional gasto para a resolução do PRVC.

## 1.3 METODOLOGIA

O presente trabalho inicia-se com uma pesquisa baseada em livros, revistas, dissertações e artigos relacionados à resolução do Problema de

Roteamento de Veículos Capacitados, ao conceito de Processamento Paralelo e a utilização de heurísticas para a resolução do PRVC.

Na sequência, são determinados e especificados os algoritmos clássicos a serem utilizados na construção da heurística proposta.

Em seguida, é viabilizada a implementação computacional da heurística proposta, utilizando técnicas de processamento paralelo, e são testadas algumas instâncias clássicas existentes na literatura.

Procurar-se-á, por fim, especificar o nível percentual médio dos desvios das soluções encontradas em relação às soluções ótimas das instâncias, validando os resultados obtidos e verificando a consistência da estratégia empregada na resolução do PRVC.

#### 1.4 ESTRUTURA

Ao término deste capítulo introdutório, será apresentado no Capítulo 2 o referencial teórico elaborado através de uma fundamentada revisão bibliográfica. Tal revisão busca apresentar, ao prezado leitor, os conceitos essenciais a respeito do Problema de Roteamento de Veículos Capacitados, do Processamento Paralelo e da utilização de heurísticas para a resolução do PRVC.

O Capítulo 3 descreve formalmente o problema estudado, apresenta a estratégia de resolução do PRVC e especifica os materiais e os métodos utilizados para a implementação computacional da heurística proposta.

O Capítulo 4 apresenta as características do *software* desenvolvido, os resultados obtidos para as instâncias e algumas considerações a cerca dos tempos de processamento computacional gastos.

O Capítulo 5 traz as conclusões presente estudo e apresenta sugestões de pesquisas para futuros trabalhos na área.



## 2 REFERENCIAL TEÓRICO

A determinação da melhor rota a ser seguida por um veículo é um instrumento valioso para a gestão de operações, sobretudo quando essas operações são inerentes à entrega de um produto ou serviço diretamente ao cliente.

Este capítulo apresenta os conceitos relacionados ao Problema de Roteamento de Veículos Capacitados, ao Processamento Computacional Paralelo e aos métodos exatos e heurísticos geralmente empregados para a resolução do PRVC.

Todos esses aspectos servirão de referência para fundamentar a estratégia utilizada na construção da nova heurística de resolução do PRVC.

### 2.1 O PROBLEMA DO ROTEAMENTO DE VEÍCULOS CAPACITADOS (PRVC)

Os problemas de construção de rotas (roteamento) são largamente estudados pela comunidade científica. Analisados como um dos típicos problemas de Pesquisa Operacional, suas implicações avançam sobre vários campos da Logística e da Gestão da Produção.

Um famoso problema de construção de rotas é o Problema do Caixeiro Viajante (PCV), onde um viajante solitário necessita determinar a melhor rota para visitar uma quantidade específica de clientes localizados em cidade diferentes. No PCV, a rota se inicia em um cliente qualquer, passa por todos os demais clientes uma única vez e regressa para o ponto inicial ao término do trajeto.

O Problema de Roteamento de Veículos (PRV) é uma variação do PCV e foi proposto inicialmente por Dantzig e Ramser (1959) para a resolução de um problema de distribuição de combustíveis em uma cidade. O PRV diferencia-se do PCV ao determinar rotas específicas para vários veículos, em oposição à utilização de um único viajante do PCV.

Variações do PRV geram inúmeras abordagens reais para os problemas das empresas, dentre os quais se destacam:

a) Problema de Roteamento de Veículos com Coleta e Entrega: nesta abordagem os produtos são retirados de um cliente e entregues em outro diferente ao longo da rota percorrida;

b) Problema de Roteamento de Veículos com Janela de Tempo: neste problema os clientes possuem janelas de tempo específicas para os veículos realizarem as entregas dos produtos;

c) Problema de Roteamento de Veículos Capacitados: nesta abordagem os veículos possuem variáveis restritivas (capacidades) como, por exemplo, a distância máxima a ser percorrida ou a carga máxima a ser transportada em cada veículo.

No presente trabalho, será analisada a abordagem do Problema de Roteamento de Veículo Capacitados para identificar um grupo de rotas ideais para uma frota homogênea de veículos, partindo de um depósito central e com destino a um conjunto específico de clientes.

Utilizando essa abordagem, cada cliente deve ser atendido por apenas um veículo e a demanda transportada não deve extrapolar a capacidade de carregamento do mesmo. Objetiva-se, por fim, determinar as rotas que minimizem a distância total percorrida pela frota de veículos.

## 2.2 A ESTRATÉGIA “*CLUSTER FIRST ROUTE SECOND*”

A estratégia “*Cluster First Route Second*” para a resolução do PRVC baseia-se na aplicação de duas fases elementares de cálculo para a obtenção da solução final. Sendo comumente reportada na literatura, essa estratégia consiste primeiramente em agrupar os clientes conforme a localização e a quantidade de produtos demandada. Na sequência, são construídas as rotas necessárias para atender a cada grupo de clientes formado, sempre considerando a capacidade máxima de carregamento do veículo.

Garantindo que a demanda dos clientes pertencentes a um grupo não exceda a capacidade do veículo, o PRVC recai na resolução do Problema do Caixeiro Viajante (PCV) ao ensejar um roteiro para cada grupo que passe por todos os pontos e apresente, ao final, a menor distância total possível.

Fisher e Jaikumar (1981) e Gillett e Johnson (1976) são exemplos de autores que apontam heurísticas do tipo “Cluster First Route Second” para a resolução do PRVC.

Variações do problema clássico tratadas com essa abordagem são encontradas em Sariklis e Powell (2000) que trabalham para a resolução do *OVRP* (*Open Vehicle Routing Problem*). O *OVRP* se diferencia do PRVC por não necessitar do retorno do veículo ao depósito ao término da rota.

A estratégia “Cluster First Route Second” também é utilizada por Steiner *et al.* (2000) para a resolução de problema de roteamento em transporte escolar. Os autores sugerem o emprego da estratégia tanto para o roteamento capacitado quanto para o roteamento não capacitado de veículos.

### 2.3 PROCESSAMENTO COMPUTACIONAL PARALELO

Leijen e Hall (2007) indicam que os computadores de vários núcleos tornaram-se o padrão industrial aplicado atualmente. Infelizmente, segundo os mesmo autores, a maioria dos aplicativos computacionais utiliza um só núcleo e não apresenta aumento de velocidade quando executada em uma máquina com vários núcleos. Dessa maneira, para aprimorar o desempenho de um *software* torna-se necessário executá-lo em vários processadores em paralelo.

Novas ferramentas de programação foram desenvolvidas para empregar os recursos de paralelismo em computadores multinucleados. Dentre elas, Leijen e Hall (2007) destaca a classe *Parallel* presente no *.NET Framework 4* e exemplificada na Figura 1 a seguir:

```
Parallel.For(0, 100, delegate (int i) {  
    a[i] = a[i]*a[i];  
});
```

Figura 1 - Utilização da classe *Parallel* em trecho de código paralelizado  
Fonte: Leijen e Hall (2007)

Na Figura acima, os autores utilizam a comando “*For*” para elevar ao quadrado os dados armazenados em uma matriz. A classe *Parallel* foi utilizada para

determinar que os cálculos necessários sejam executados de forma simultânea pelo aplicativo.

Essa determinação torna-se viável uma vez que as iterações são independentes entre si, ou seja, as novas iterações não necessitam ler as atualizações de dados feitas pelas iterações anteriores.

Neste trabalho utiliza-se a linguagem de programação C# 4.0 em conjunto com a extensão de linguagem denominada *Language Integrated Query (LINQ)*, além da classe *Parallel*. A LINQ possibilita uma manipulação de conjuntos de dados e a classe *Parallel* permite o devido processamento paralelo desses dados pelo *software*.

Embora represente uma grande evolução em termos de programação computacional, Toub (2011) reforça que a classe *Parallel* só chega a abordar superficialmente as funcionalidades disponíveis para paralelismo.

Conforme o mesmo autor, um dos mais formidáveis avanços em paralelização foi o realizado pelo *.NET Framework 4* através da introdução da extensão *Parallel LINQ (PLINQ)*. Essa extensão permite o processamento paralelo dos dados internos de um conjunto, desde que os mesmos possuam alto grau de independência e haja outros núcleos de processamento disponíveis.

A Figura 2 abaixo exemplifica a utilização da extensão *PLINQ* – *AsParallel()* – dentro do *software* desenvolvido neste trabalho:

```
var medianaProxima = (from m in mMedianas.AsParallel()
                      orderby m % designando.cliente.Coordenada
                      where m.CapacidadeDisponivel >= designando.cliente.Demanda
                      select m).First();
```

Figura 2 - Exemplo de aplicação de PLINQ no *software* criado.  
Fonte: Elaborado pelo autor.

O trecho de código exposto acima visa ordenar dados para selecionar uma próxima mediana dentro do grupo denominado “mMedianas”. Esse cálculo pode ser executado e processado de maneira paralela pela sua independência em relação à iteração subsequente.

O diferencial do processamento computacional paralelo é, portanto, a realização de cálculos matemáticos em núcleos de processamento distintos, porém simultaneamente, através de capacidades atribuídas pelo programador ao *software*.

Visando reduzir o tempo de processamento e aumentar a capacidade de análise do *software*, a heurística proposta nesse trabalho bem como os algoritmos utilizados na sua implementação foram programados utilizando os conceitos de processamento paralelo supracitados.

Cabe destacar, por fim, que somente a execução dos cálculos matemáticos pelo *software* foi paralelizada e não as sequências de passos preconizadas nos algoritmos da heurística proposta.

## 2.4 ESTRATÉGIAS PARA RESOLUÇÃO DO PRVC

Na literatura existem diversas abordagens para a resolução do PRVC, entretanto todas elas podem ser agrupadas, de modo geral, em duas classes distintas: métodos exatos e métodos heurísticos.

Quanto aos métodos exatos, alguns trabalhos científicos são baseados em algoritmos *branch-and-cut* ou relaxação lagrangeana. Christofides *et al.* (1981) apresenta um dos trabalhos mais impactantes para a utilização dos métodos exatos. Nessa ocasião, os autores empregaram limitantes lagrangeanos para a geração das subrotas dos veículos. O algoritmo *branch-and-bound* desenvolvido foi capaz de resolver instâncias com até 25 clientes.

Outro importante trabalho é o apresentado por Fukasawa *et al.* (2006) onde o um algoritmo *branch-and-cut-and-price* obteve resultados ótimos para todas as instâncias com mais de 135 clientes.

Trabalhos científicos recentes intensificam o emprego de planos de corte, como se verifica em Achuthan *et al.* (2003). Nesse trabalho, os autores utilizaram novos planos de corte baseados em um algoritmo *branch-and-cut* e testaram sobre instâncias cujas faixas de pontos variavam entre 15 e 100 clientes, além de 24 problemas clássicos da literatura. O trabalho de Achuthan *et al.* (2003) produziu relevantes ganhos em termos de tempo computacional.

No que tange aos métodos heurísticos, Campos e Mota (2000) apresentaram duas estratégias heurísticas. A primeira é baseada em *scratch* que gera uma solução inicial sem qualquer informação obtida *a priori*. A segunda emprega informações advindas de relaxações lineares fortes a partir do problema original.

Em ambas as situações, os autores utilizaram técnicas de busca tabu para refinar as soluções iniciais e testaram as heurísticas em instâncias cujas faixas de pontos variavam entre 22 e 135 clientes. Cabe ressaltar que os autores obtiveram as soluções ótimas para a maioria delas.

Utilizando também métodos heurísticos, Zhishuo e Yueting (2005) desenvolveram um novo método baseado em colônia de formigas (*Ant Colonies Algorithm*). O trabalho apresenta um procedimento para a construção de sub-rotas iniciando a partir dos vértices mais distantes, escolhidos randomicamente. Essa abordagem é oposta ao tradicional método de se iniciar a construção de uma sub-rota pelo depósito tornando-se, portanto, uma inovação.

Zhishuo e Yueting (2005) ainda empregam um algoritmo de varredura sobre as sub-rotas da solução atual, trocando pares de vértices entre elas. Os resultados obtidos foram apenas 0,28% inferiores aos valores ótimos estabelecidos nas instâncias testadas. Tal fato reforça o desempenho superior desta estratégia frente às demais abordagens baseadas em colônia de formigas para o PRVC.

Pelo exposto nos parágrafos acima, fica evidente a quantidade de estratégias diferentes de resolução do PRVC passíveis de serem implementadas. O presente trabalho opta por incrementar o rol dessas estratégias ao sugerir a construção de uma nova abordagem heurística, mais simples e cuja implementação computacional produza tempos de processamento reduzidos.

### 3 MATERIAIS E MÉTODOS

Este capítulo irá descrever a problemática estudada, os algoritmos matemáticos empregados e a estratégia utilizada para a resolução do PRVC que culminará, ao final, na heurística proposta.

#### 3.1 DESCRIÇÃO FORMAL DO PRVC

Guimarães, Scarpin e Steiner (2011) expõem que o PRVC pode ser formalmente definido sob o seguinte aspecto: seja  $G(V, E)$  um grafo não direcionado contendo o conjunto de vértices  $V = \{0, 1, \dots, n\}$ , onde o vértice “0” representa o depósito, enquanto que todos os outros vértices representam os clientes, sendo que cada cliente  $i$  possui uma demanda  $d_i$  associada. Cada arco  $e \in E$  possui um comprimento não negativo  $\ell(e)$ . Dado  $G$  e dois números positivos e inteiros ( $K$  e  $Q$ ), o PRVC consiste em encontrar um conjunto de rotas para os  $K$  veículos, que atenda as restrições:

- (i) Cada rota inicia e termina no depósito;
- (ii) Cada cliente é visitado por um único veículo;
- (iii) A demanda total de todos os clientes de uma sub-rota não excede a capacidade  $Q$  do veículo.

A Figura 3 a seguir apresenta um exemplo de uma instância de cálculo:

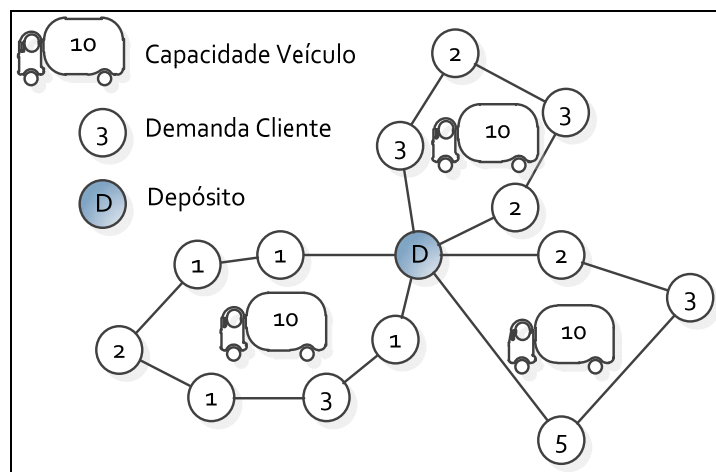


Figura 3 - Instância exemplo do PRVC.  
Fonte: Elaborado pelo Autor

Ainda segundo Guimarães, Scarpin e Steiner (2011), o objetivo do PRVC é:

“[...] minimizar o somatório das distâncias de todas as sub-rotas determinadas. O problema é fortemente NP-hard, dado que é uma generalização do Problema do Caixeiro Viajante (PCV) demandando, portanto, elevado esforço computacional para obtenção da solução ótima através de busca exaustiva.”

Baseando-se em Lin *et al.* (2009), o PRVC pode ser modelado como um problema de Programação Inteira Misto conforme indicado a seguir:

$$\text{Minimizar: } \sum_{i=0}^N \sum_{j=0}^N \sum_{k=1}^K C_{ij} X_{ijk} \quad (1)$$

$$\text{Sujeito à: } \sum_{i=0}^N \sum_{j=0}^N X_{ijk} d_i \leq Q^k, \quad 1 \leq k \leq K, \quad (2)$$

$$\sum_{j=1}^N X_{ijk} = \sum_{j=1}^N X_{jik} \leq 1, \quad \text{para } i = 0 \text{ e } k \in \{1, \dots, K\}, \quad (3)$$

$$\sum_{k=1}^K \sum_{j=1}^N X_{ijk} \leq K, \quad \text{para } i = 0, \quad (4)$$

onde  $C_{ij}$  é o custo para ir do cliente  $i$  para o cliente  $j$ ,  $K$  é o número de veículos disponível,  $N$  é o número de clientes,  $Q^k$  é a capacidade de carregamento do veículo  $k$ ,  $d_i$  é a demanda do cliente  $i$ . As variáveis  $X_{ijk} \in \{0\}$  ou  $\{1, (i \neq j)\}$   $\forall i, j \in \{0, 1, \dots, N\}$ .

A função objetivo do problema é minimizar o somatório das distâncias de todas as sub-rotas e está representada pela equação (1). As restrições (2) referem-se à capacidade de carregamento do veículo e ao atendimento das demandas de cada cliente, onde  $X_{ijk} = 1$ , se o veículo  $k$  viaja do cliente  $i$  para o cliente  $j$  diretamente e  $X_{ijk} = 0$ , caso contrário.

A garantia de que toda rota tem início e fim no depósito é imposta pelas restrições (3), enquanto que a restrição (4) especifica que há no máximo  $K$  rotas partindo do depósito.



Pela complexidade do equacionamento exposto, métodos exatos de cálculo tornam-se difíceis de serem aplicados. O presente trabalho enseja à utilização de métodos heurísticos como pode ser observado nos itens 3.2 e 3.3 a seguir.

## 3.2 ESTRATÉGIA DE RESOLUÇÃO

Conforme mencionado anteriormente, problemas de grande porte como o PRVC desmotivam o emprego de modelos exatos de programação linear. Essa situação surge pelo fato do tempo de processamento aumentar exponencialmente com o incremento do número de clientes (nós) e de veículos.

Devido a isto, métodos heurísticos vêm sendo propostos para a resolução do PRVC no intuito de se produzir bons resultados com tempos de processamento factíveis e adequados as necessidade empresariais.

A construção da heurística proposta neste trabalho requer a resolução sequencial dos problemas da  $p$ -medianas, da designação ou agrupamento (clusterização), da roteirização de veículos e do refinamento de rotas.

Para uma melhor compreensão por parte do leitor, as subseções a seguir discutem esses problemas de forma pormenorizada para que a nova abordagem heurística seja devidamente apresentada ao final.

### 3.2.1 O problema das $p$ -medianas

O problema das  $p$ -medianas tem por objetivo determinar, em um grafo formado por um conjunto de pontos (clientes),  $p$ -pontos principais que representam sementes ou depósitos fictícios.

Os demais pontos (clientes) serão designados a cada uma destas medianas de modo a formar  $p$ -conjuntos iniciais ( $p$ -grupos) de clientes onde cada conjunto contenha a “capacidade limite” igual à capacidade de carregamento dos veículos.

No presente trabalho a capacidade do grupo de clientes é a mesma para todos eles uma vez que a análise será realizada considerando-se uma frota homogênea de veículos.

A estratégia de resolução reside na teoria dos grafos que tem por objetivo localizar facilidades ao longo de uma rede viária. Estas facilidades ( $p$ -medianas) devem ser escolhidas de forma a minimizar a soma da distância de cada um dos pontos à facilidade mais próxima, ponderada por um fator de demanda.

O problema pode ser resolvido de forma exata, como sugeri Hakimi (1965), usando enumeração exaustiva ou programação inteira. Todavia, para problemas de grande porte (grande quantidade de pontos), torna-se necessário recorrer a métodos aproximados (heurísticos) tendo em vista o esforço computacional requerido pelos métodos exatos ser muito elevado.

Dentre os métodos heurísticos disponíveis, destaca-se o algoritmo proposto por Teitz e Bart (1968) pela sua eficiência, estabilidade e poder de generalização. Tal algoritmo é detalhadamente descrito pela Figura 4 a seguir:

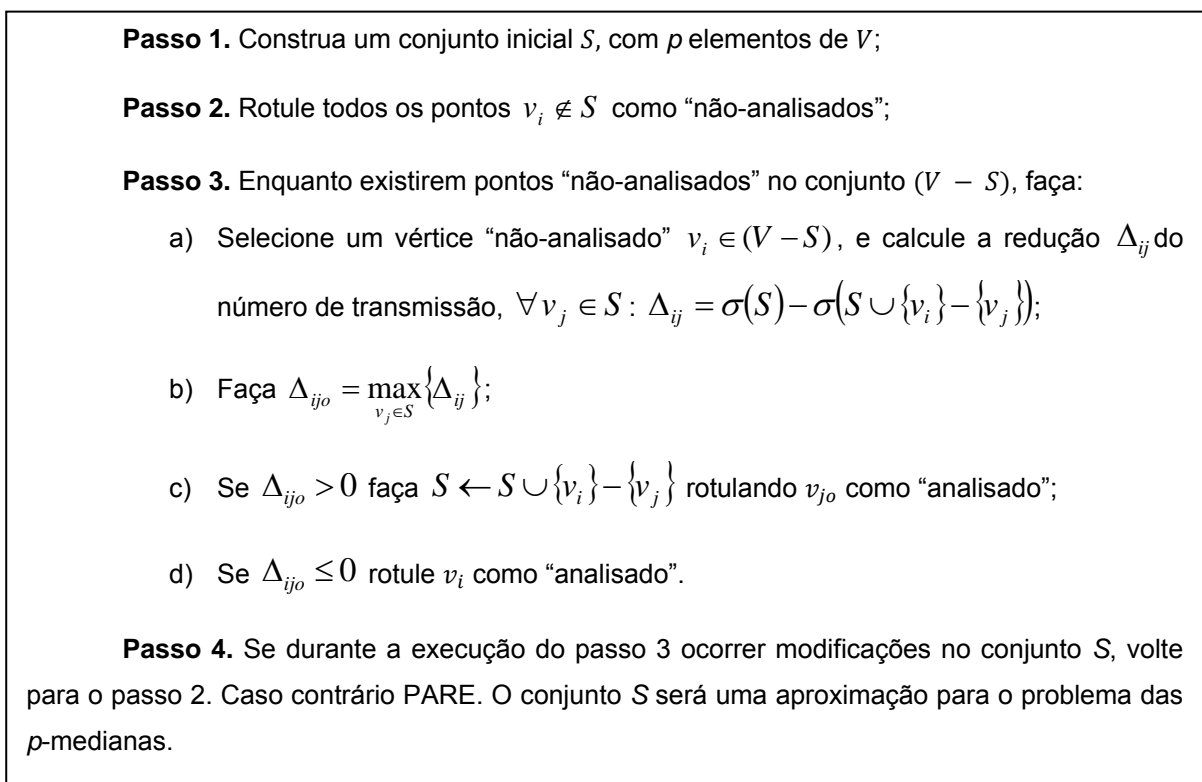


Figura 4 - Heurística de Teitz e Bart.  
Fonte: Teitz e Bart (1968)

### 3.2.2 O problema da formação de grupos de clientes (clusterização)

Definido o conjunto das medianas é necessário designar os pontos de maneira a se compor o agrupamento para roteirização posterior. Esta designação objetiva minimizar a distância entre os pontos do conjunto e a respectiva mediana, respeitando a capacidade da própria mediana.

Neste trabalho, a designação dos pontos é feita através do algoritmo de Gillett e Johnson (1976). O princípio do algoritmo é construir uma lista ordenada dos pontos a serem designados considerando distâncias entre esses pontos e as duas medianas mais próximas.

O algoritmo é descrito pela Figura 5 a seguir:

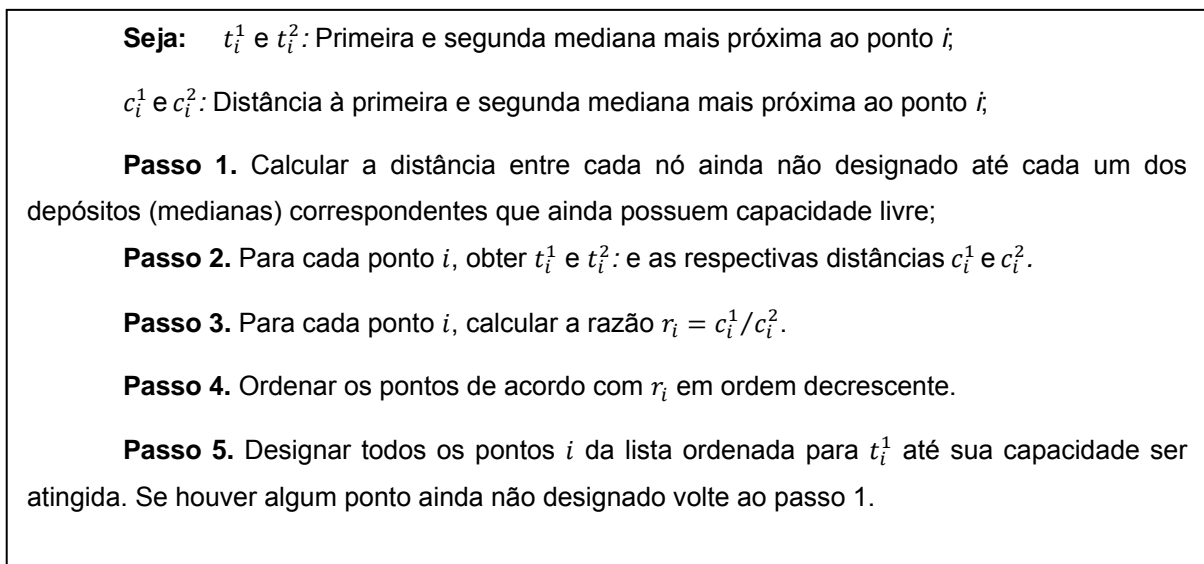


Figura 5 - Algoritmo de Gillett e Johnson para agrupamento.  
 Fonte: Gillett e Johnson (1976)

### 3.2.3 O problema da roteirização: construção de uma solução Inicial

Inúmeros são os algoritmos existentes para a roteirização de veículos tendo um conjunto de clientes já definido. A resolução do problema do roteamento enseja a obtenção de uma solução inicial para posterior refinamento. Para a abordagem proposta neste trabalho foi utilizado o algoritmo do Vizinheiro Mais Próximo (VMP) pela sua ampla divulgação na literatura existente e fácil implementação computacional.

O algoritmo de roteirização VMP foi originalmente proposto por Cover e Hart (1967) e consiste na composição da rota baseada em uma inserção sequencial de pontos, através de um ponto inicial, conforme a menor distância entre este e os seus demais pontos (vizinhos).

Após determinar todas as distâncias entre o ponto inicial e os demais pontos do cluster, os demais pontos são ordenados de maneira decrescente para possibilitar a escolha e determinação do nó mais próximo que será designado à rota.

O algoritmo está exposto na Figura 6 a seguir:

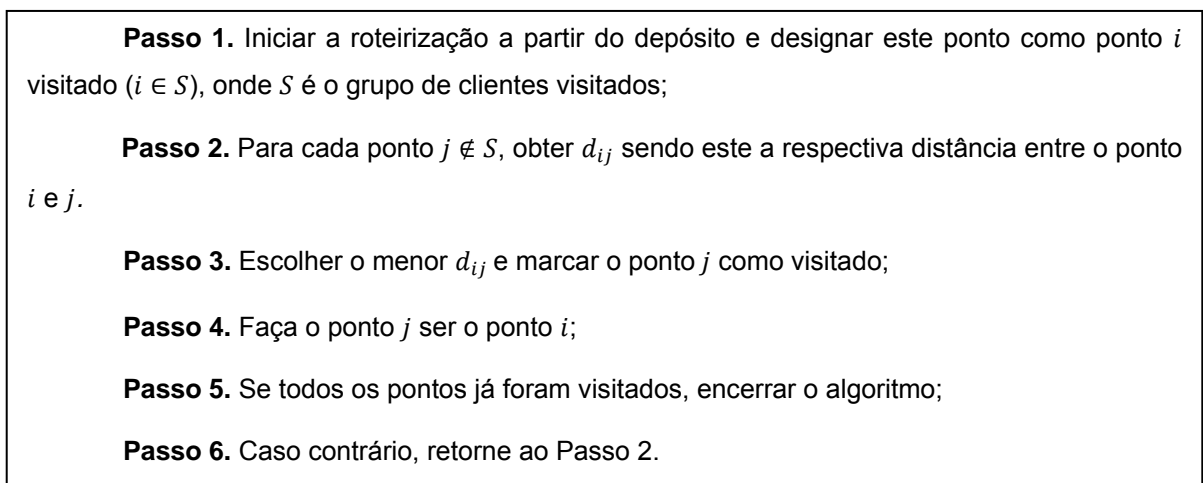


Figura 6 - Algoritmo do Vizinho Mais Próximo (VMP).  
Fonte: Cover e Hart (1967)

### 3.2.4 O problema da roteirização: refinamento da solução inicial

O problema da melhoria de rotas busca aumentar a eficiência do trajeto, ou seja, refinar uma solução inicial obtida para uma rota a ser seguida, *a priori*, por um veículo. Uma das melhores abordagens heurísticas para a resolução desse problema é a proposta por Lin e Kernigham (1973) denominada troca de arcos  $k$ -opt, onde as trocas 2-opt (2 arcos) e 3-opt (3 arcos) são as mais utilizadas.

Os métodos  $k$ -opt buscam a melhoria de um trajeto pela substituição de " $k$ " arcos no roteiro estabelecido anteriormente, isto é, " $k$ " arcos são removidos do roteiro e substituídos por outros " $k$ " arcos. Caso alguma melhoria seja detectada, a troca é aceita e o novo arco passa a compor a solução incumbente. Neste trabalho esta dinâmica se repete até que nenhuma troca outra troca resulte em melhoria.

Conforme apontado por Laporte *et al.* (1999), o processo de melhoria  $k$ -opt termina em um mínimo local e possui ordem de complexidade  $O(nk)$ . Quanto maior for o valor de  $k$ , melhor será a solução, entretanto o esforço computacional requerido também será maior. Isto leva a um *trade-off* entre qualidade e tempo computacional. Dessa forma, trocas 3-opt e superiores ensejam um custo computacional muitas vezes superior à melhoria da solução obtida.

Neste trabalho optou-se por utilizar a troca 2-opt pela sua simplicidade de implementação computacional, uma vez que esta estratégia de melhoria realiza a comparação somente entre dois trechos para então determinar se a inversão de sentido entre eles é mais econômica em relação à rota inicial. Caso seja, a nova rota passa a ser a rota preferencial.

Na Figura 7 abaixo, é ilustrada a sistemática de troca de arcos para a abordagem 2-opt que será utilizada na heurística proposta no trabalho:

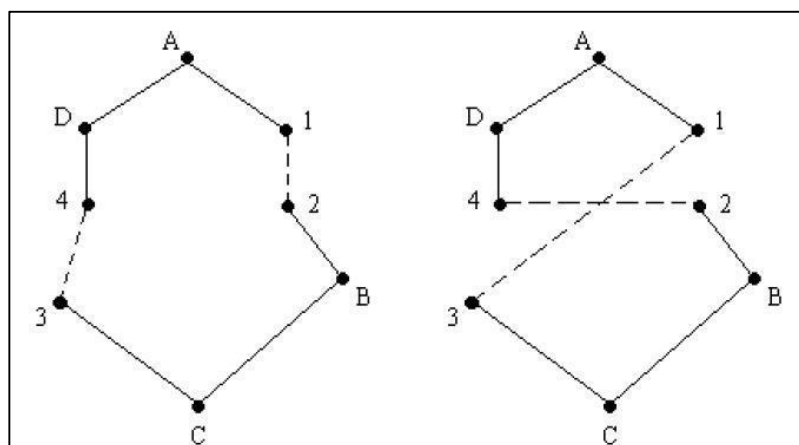


Figura 7 - Troca 2-opt entre os trechos 1-2 e 3-4  
Fonte: Adaptado de Costa (1997)

Cabe destacar que a geração de boas soluções iniciais é fundamental para que se obtenham boas soluções finais com os métodos  $k$ -opt para a melhoria de rotas (COSTA, 1997; CAMPOS, 2008).

### 3.3 HEURÍSTICA PROPOSTA PARA A RESOLUÇÃO DO PRVC

A heurística proposta neste trabalho foi desenvolvida e implementada computacionalmente seguindo a concatenação de algoritmos representada através da Figura 8 a seguir:

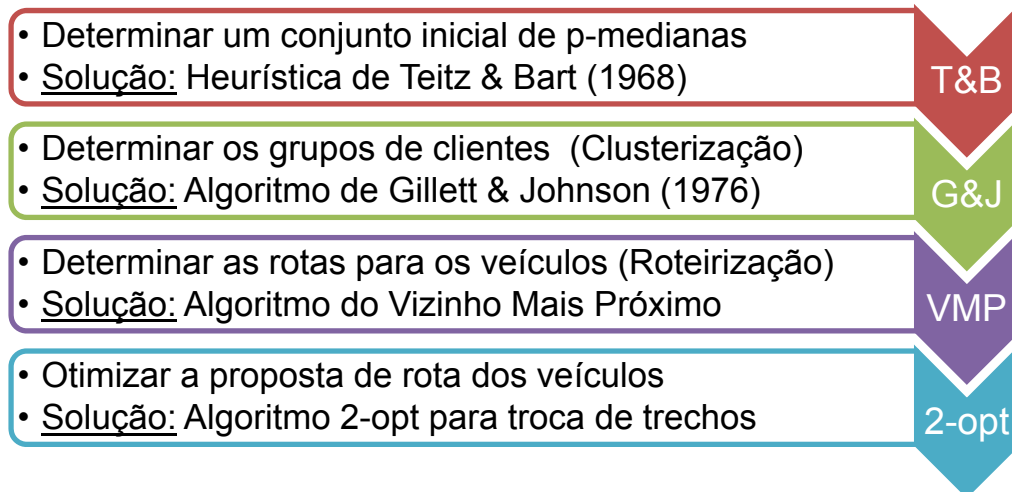


Figura 8 - Heurística proposta para a resolução do PRVC  
Fonte: Elaborado pelo autor

Para o melhor entendimento do funcionamento da heurística proposta, a Figura 9 a seguir apresenta o pseudocódigo desenvolvido e implementado computacionalmente:

	<b>Procedimento</b> Rotina ( <i>instância</i> )
1	<b>Início</b>
2	<i>iteração</i> ← 0
3	<b>Enquanto</b> <i>iteração</i> < 100 <b>faz</b>
4	<i>SoluçãoInicial</i> ← ProcessarTeitzEBartComGillettJohnson( <i>instância</i> )
5	<i>Roteirização</i> ← ProcessarVizinhoMaisProximo( <i>SoluçãoInicial</i> , <i>instância</i> )
6	<i>Resposta</i> ← 2-Opt( <i>Roteirização</i> , <i>instância</i> )
7	<i>iteração</i> ← <i>iteração</i> + 1
8	<b>Se</b> <i>Resposta.CustoTotal</i> < <i>Solução.CustoTotal</i> <b>então</b>
9	<i>Solução</i> ← <i>Resposta</i>
10	<b>FimSe</b>
11	<b>FimEnquanto</b>
12	<b>Retorna</b> <i>Solução</i>
13	<b>Fim</b>

Figura 9 - Pseudocódigo para a heurística proposta  
Fonte: Elaborado pelo autor

Visando facilitar o gerenciamento e a elaboração do código fonte do *software*, utilizou-se o ambiente colaborativo disponibilizado pelo *website* [www.codeplex.com](http://www.codeplex.com).

Nesse sentido, todo o código fonte encontra-se disponível em <http://prvc.codeplex.com>, enfatizando a natureza do *software* (*open source*) e

possibilitando a proposição de melhorias por outros programadores interessados no assunto.

Quanto ao *hardware* utilizado, o *software* supracitado foi executado em um processador Intel® Core™2 Quad Q6600, 2.4 GHz com 4 GB de RAM e sistema operacional Windows 7™, 64-Bits.

A heurística proposta foi avaliada sobre as instâncias clássicas da literatura disponíveis no *website* <http://www.branchandcut.org>. Cabe destacar que foram avaliadas somente as instâncias cujos dados disponíveis estavam compreendidos no espaço vetorial de domínio Euclidiano (2D).

As instâncias classificadas como não-euclidianas diferenciam-se das euclidianas por pré-definirem uma matriz de dados contendo todas as distâncias possíveis e existentes entre todos os seus pontos (clientes).

Em grande parte dessas instâncias o sentido de percurso é relevante e implica em distâncias diferentes para os trechos de ida e volta entre dois pontos. Por esse motivo, as instâncias não-euclidianas não fazem parte ao escopo deste trabalho.

Dessa forma, foram avaliadas as 83 instâncias euclidianas pertencentes às classes A, B, E, G, M, P, V e evidenciadas na Tabela 1 abaixo:

Tabela 1 - Instâncias utilizadas para avaliação da heurística proposta

<b>Instância</b>	<b>Instância</b>	<b>Instância</b>	<b>Instância</b>	<b>Instância</b>	<b>Instância</b>
A-n32-k5	A-n53-k7	B-n38-k6	B-n68-k9	M-n121-k7	P-n55-k8
A-n33-k5	A-n54-k7	B-n39-k5	B-n78-k10	M-n151-k12	P-n60-k10
A-n33-k6	A-n55-k9	B-n41-k6	E-n101-k14	M-n200-k17	P-n60-k15
A-n34-k5	A-n60-k9	B-n43-k6	E-n101-k8	P-n101-k4	P-n65-k10
A-n36-k5	A-n61-k9	B-n44-k7	E-n22-k4	P-n16-k8	P-n70-k10
A-n37-k5	A-n62-k8	B-n45-k5	E-n23-k3	P-n20-k2	P-n76-k4
A-n37-k6	A-n63-k10	B-n50-k7	E-n30-k3	P-n21-k2	P-n76-k5
A-n38-k5	A-n63-k9	B-n50-k8	E-n33-k4	P-n22-k2	att-n48-k4
A-n39-k5	A-n64-k9	B-n52-k7	E-n51-k5	P-n40-k5	
A-n39-k6	A-n65-k9	B-n56-k7	E-n76-k10	P-n45-k5	
A-n44-k6	A-n69-k9	B-n57-k9	E-n76-k14	P-n50-k10	
A-n45-k6	A-n80-k10	B-n63-k10	E-n76-k7	P-n50-k7	
A-n45-k7	B-n31-k5	B-n64-k9	E-n76-k8	P-n51-k10	
A-n46-k7	B-n34-k5	B-n66-k9	G-n262-k25	P-n55-k10	
A-n48-k7	B-n35-k5	B-n67-k10	M-n101-k10	P-n55-k7	

Fonte: Elaborado pelo autor

Cabe ressaltar, por fim, que o critério diferenciador das classes de instâncias euclidianas pode variar desde a região onde os clientes estão posicionados até a densidade dos mesmos em relação ao plano 2D.



## 4 RESULTADOS E DISCUSSÕES

### 4.1 O SOFTWARE DESENVOLVIDO

O *software* desenvolvido para a análise das instâncias possui a interface do usuário (UI) indicada na Figura 10 abaixo:

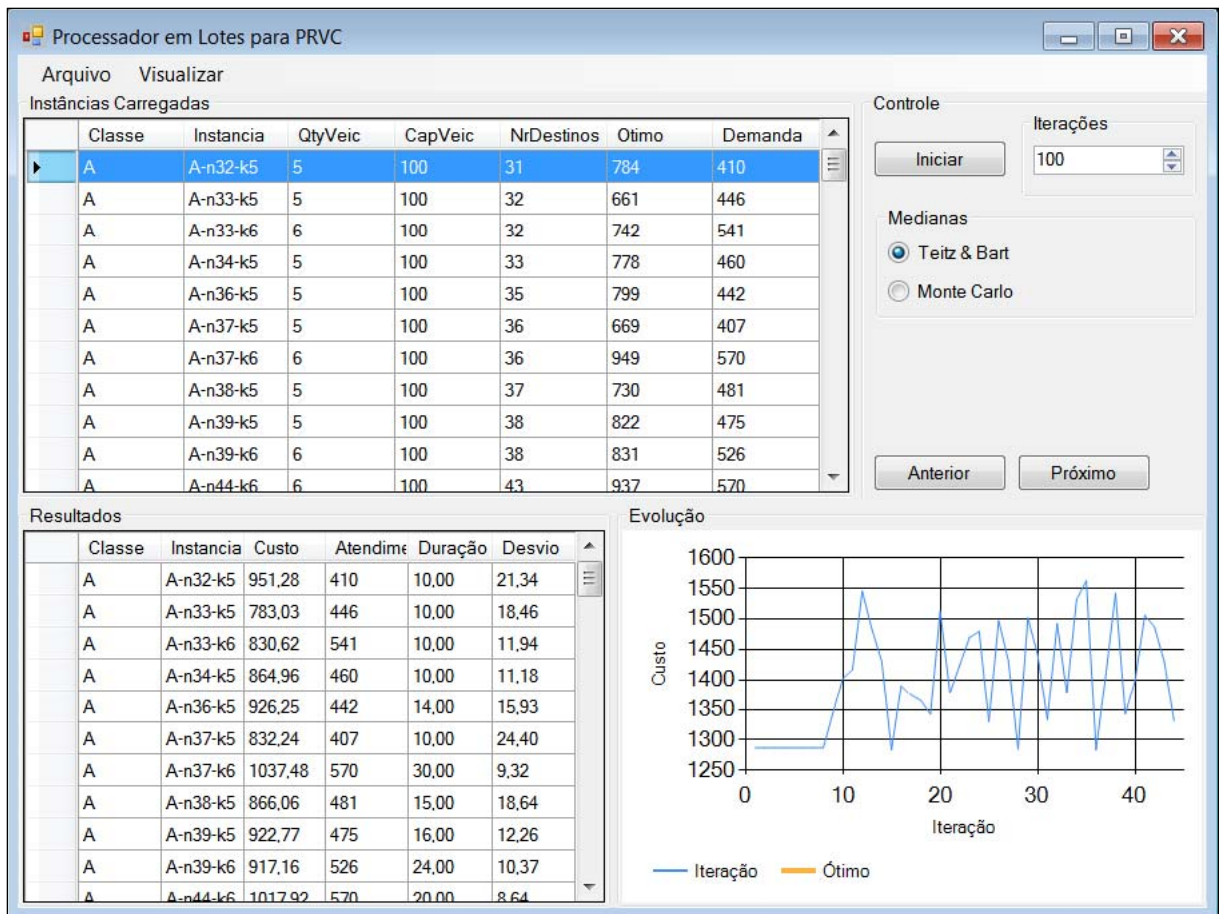


Figura 10 - Interface do usuário para o software desenvolvido

Fonte: Elaborado pelo autor

Essa interface apresenta basicamente quatro grupos de informações: Instâncias Carregadas, Controle, Resultados e Evolução.

No canto superior esquerdo da UI (Instâncias Carregadas) são elencadas as instâncias para a realização dos testes, juntamente com as informações principais de cada instância. Da esquerda para a direita, temos: a classe e o nome da instância, a quantidade de veículos, o número de destinos atendidos, a solução ótima e a sua demanda total.

Na parte inferior esquerda da UI (Resultados), temos as seguintes informações: a classe e o nome da instância, o caminho total da solução ótima encontrada (Custo), a demanda atendida (Atendimento), o tempo de processamento (Duração) e o desvio da solução obtida em relação à solução ótima da instância. Cabe destacar que o tempo de processamento está exposto em milissegundos (ms) e o desvio em porcentagem (%).

No canto superior direito da UI (Controle) é possível especificar a quantidade de iterações a serem executadas e o método de resolução do problema das  $p$ -medianas. Após escolher na UI (Instâncias Carregadas) a instância que será resolvida, basta clicar no botão “iniciar” e o *software* calcula o resultado.

Para acompanhar o andamento da resolução da instância, foi criado no canto inferior direito da UI um gráfico Custo x Iterações (Evolução). Nessa parte da UI é possível identificar a melhoria dos resultados obtidos à medida que as iterações são executadas.

A evolução dos resultados das iterações é indicada por uma linha na cor azul e a evolução das soluções ótimas obtidas é indicada por uma linha na cor laranja. Essa diagramação é exemplificada pela Figura 11 a seguir:

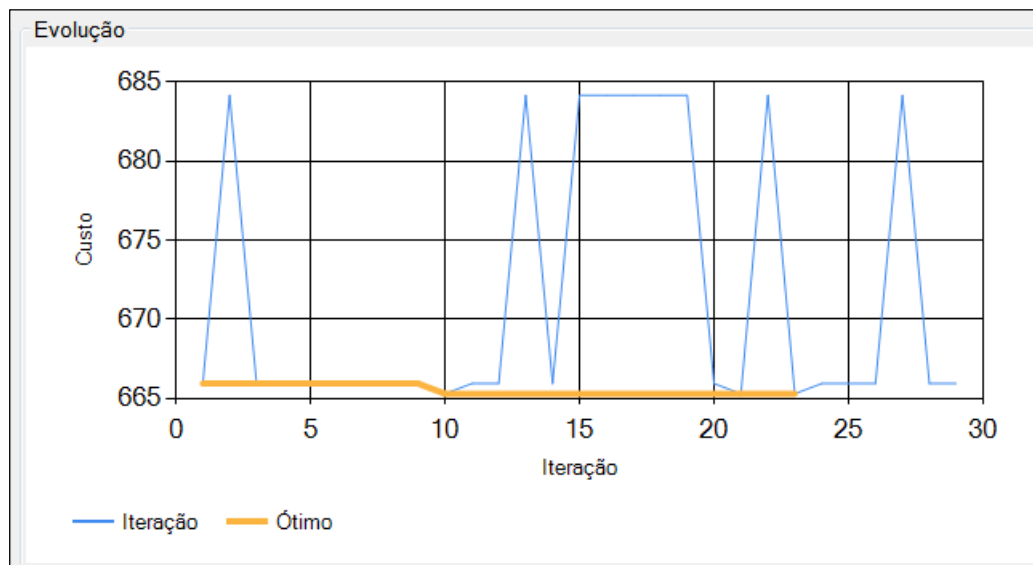


Figura 11 - Exemplo do gráfico evolutivo Custo x Iteração  
Fonte: Elaborado pelo autor

Como mencionado no Capítulo 2, técnicas de processamento paralelo foram inseridas na arquitetura do software para otimizar os cálculos e possibilitar a utilização de toda a infraestrutura computacional existente. Uma evidência do uso

dessas técnicas de programação, além dos baixos tempos de processamento, pode ser identificada quando se analisa a taxa de ocupação dos núcleos de processadores existentes.

Durante a execução do software, foram obtidos taxas de utilização da capacidade de processamento dos núcleos computacionais de 90%, conforme pode ser verificado pelo “Histórico do uso de CPU” apontado na Figura 12 a seguir:

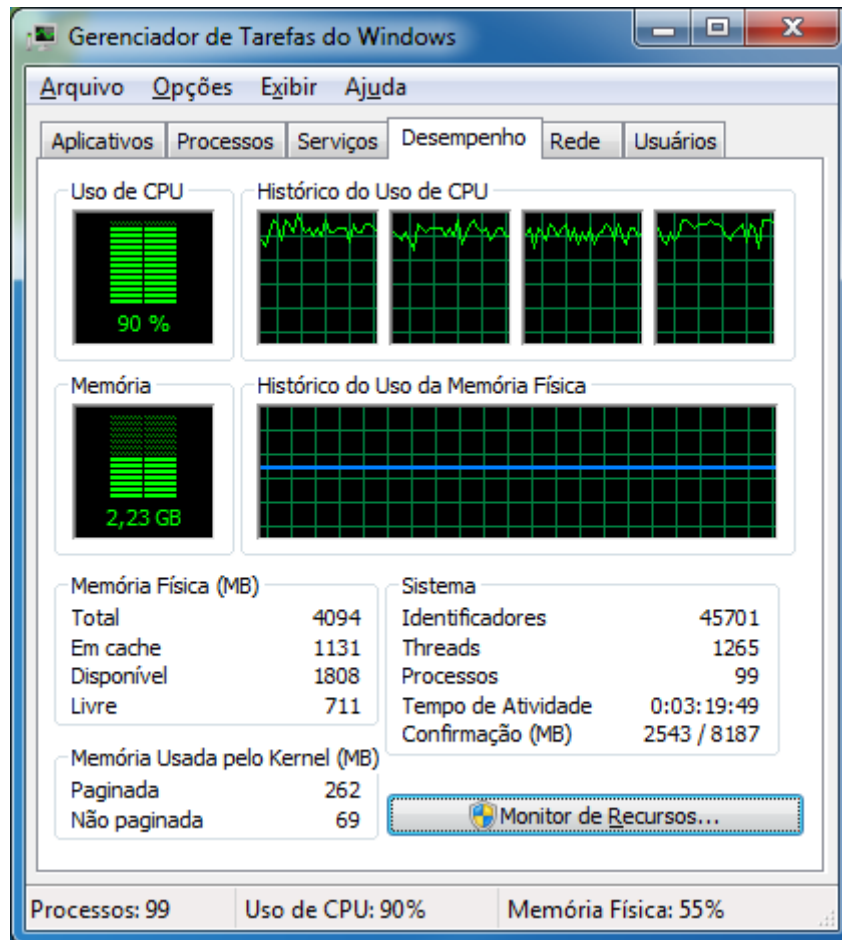


Figura 12 - Taxa de ocupação dos núcleos de processamento computacional  
Fonte: Elaborado pelo autor

O software desenvolvido ainda apresenta duas possibilidades para visualizar as rotas obtidas para cada veículo. A primeira apresenta as rotas dos veículos em formato de tabela, através da opção “Visualizar” (canto superior direito) e depois “Rotas em forma de tabela”.

Um exemplo da tabela gerada através dessa opção é exposto pela Figura 13, onde o ponto 1 representa o depósito central da instância:

	Classe	Instancia	Caminhao	Rota
▶	A	A-n32-k5	1	1, 31, 27, 17, 13, 2, 8, 1
	A	A-n32-k5	2	1, 14, 22, 32, 20, 18, 1
	A	A-n32-k5	3	1, 6, 30, 26, 11, 16, 10, 1
	A	A-n32-k5	4	1, 4, 3, 24, 29, 9, 19, 23, 12, 5, 1
	A	A-n32-k5	5	1, 21, 28, 25, 15, 7, 1

Figura 13 - Exemplo de tabela com as rotas dos veículos  
Fonte: Elaborado pelo autor

A segunda opção para a visualização das rotas obtidas para os veículos é através de um gráfico 2D gerado pelo *software*. Para tanto, basta clicar duas vezes sobre o nome da instância resolvida no canto inferior esquerdo da UI (Resultados).

Um exemplo desse gráfico é exemplificado na Figura 14 a seguir:

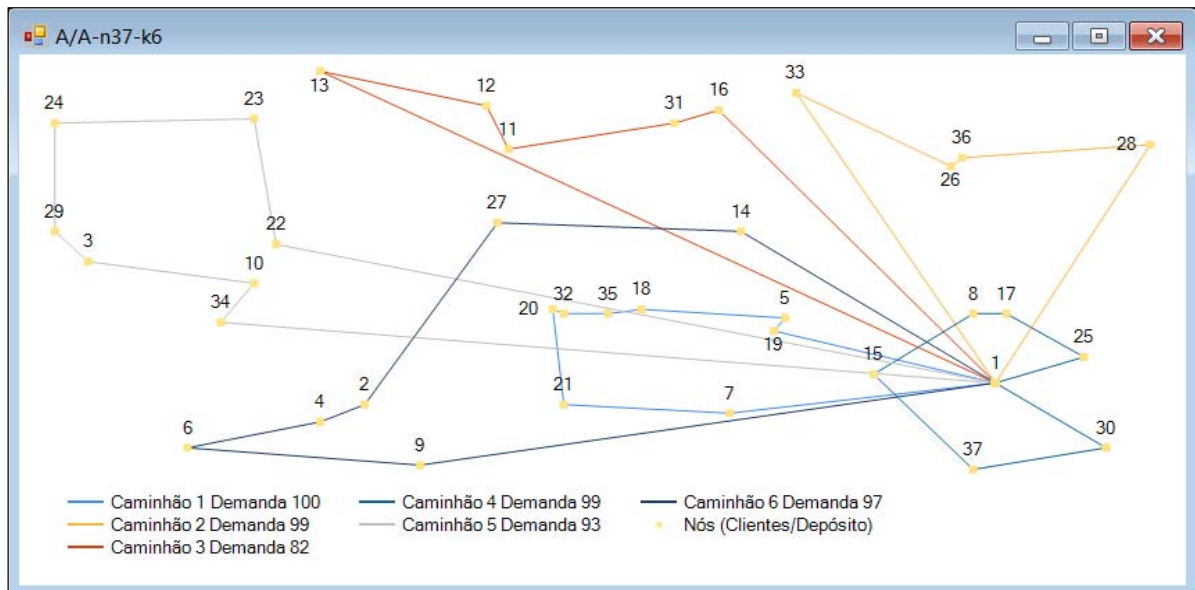


Figura 14 - Exemplo de solução gráfica 2D para as rotas dos veículos  
Fonte: Elaborado pelo autor

## 4.2 RESULTADOS OBTIDOS PARA AS INSTÂNCIAS

Todas as 83 instâncias contidas na Tabela 1 do item 3.3 deste trabalho foram devidamente resolvidas pelo *software*. Devido a grande quantidade de dados, as soluções encontradas para as rotas a serem seguidas pelos veículos estão identificadas e representadas no Apêndice A.

Nesta parte do trabalho pretende-se apenas apontar os resultados globais obtidos para as instâncias, os seus desvios percentuais em relação à solução ótima e os respectivos tempos de processamento com e sem o recurso do paralelismo computacional (*Parallel*). Para tanto, os resultados foram divididos em tabelas e apresentados na ordem alfabética das instâncias (A, B, E, G, M, P e V).

Ressalta-se que o número de pontos (clientes e depósito) está indicado no nome da própria instância após a letra “n” e o número de veículos utilizados está indicado após a letra “k”. Cabe ainda destacar que cada instância evidenciada na Tabela 1 foi resolvida pelo *software* estipulando-se um limite de 100 iterações.

Para as instâncias da classe A, foram obtidos os resultados constantes da Tabela 2 abaixo:

Tabela 2 - Resultados para as instâncias da classe A

<b>Instância</b>	<b>Número de Pontos</b>	<b>Solução Ótima</b>	<b>Solução Obtida</b>	<b>Desvio da Solução (%)</b>	<b>Tempo com <i>Parallel</i> (ms)</b>	<b>Tempo sem <i>Parallel</i> (ms)</b>
A-n32-k5	32	784	951,3	21,3	10	16
A-n33-k5	33	661	783,0	18,5	10	16
A-n33-k6	33	742	830,6	11,9	10	16
A-n34-k5	34	778	865,0	11,2	10	16
A-n36-k5	36	799	926,2	15,9	14	16
A-n37-k5	37	669	832,2	24,4	10	16
A-n37-k6	37	949	1037,5	9,3	30	31
A-n38-k5	38	730	866,1	18,6	15	16
A-n39-k5	39	822	922,8	12,3	16	31
A-n39-k6	39	831	917,2	10,4	24	47
A-n44-k6	44	937	1017,9	8,6	20	47
A-n45-k6	45	944	1068,4	13,2	30	47
A-n45-k7	45	1146	1329,8	16,0	44	62
A-n46-k7	46	914	1067,3	16,8	40	94
A-n48-k7	48	1073	1203,7	12,2	45	94
A-n53-k7	53	1010	1302,7	29,0	98	109
A-n54-k7	54	1167	1407,4	20,6	60	140
A-n55-k9	55	1073	1287,1	20,0	90	125
A-n60-k9	60	1354	1624,6	20,0	128	374
A-n61-k9	61	1034	1351,2	30,7	110	328
A-n62-k8	62	1288	1436,3	11,5	133	218
A-n63-k9	63	1616	1935,2	19,8	160	296
A-n63-k10	63	1314	1485,8	13,1	180	421
A-n64-k9	64	1401	1694,2	20,9	156	421
A-n65-k9	65	1174	1456,1	24,0	168	437
A-n69-k9	69	1159	1417,7	22,3	146	499
A-n80-k10	80	1763	2020,0	14,6	267	920
			<b>Média</b>	<b>17,3</b>	<b>75</b>	<b>180</b>

Fonte: Elaborado pelo autor

Para as instâncias da classe B, foram obtidos os resultados constantes da Tabela 3 a seguir:

Tabela 3 - Resultados para as instâncias da classe B

<b>Instância</b>	<b>Número de Pontos</b>	<b>Solução Ótima</b>	<b>Solução Obtida</b>	<b>Desvio da Solução (%)</b>	<b>Tempo com Parallel (ms)</b>	<b>Tempo sem Parallel (ms)</b>
B-n31-k5	31	672	694,1	3,3	10	16
B-n34-k5	34	788	940,9	19,4	10	16
B-n35-k5	35	955	1053,4	10,3	10	16
B-n38-k6	38	805	864,5	7,4	26	31
B-n39-k5	39	549	696,6	26,9	10	16
B-n41-k6	41	829	1071,7	29,3	20	47
B-n43-k6	43	742	832,2	12,2	20	47
B-n44-k7	44	909	989,5	8,9	50	62
B-n45-k5	45	751	929,8	23,8	30	47
B-n50-k7	50	741	865,3	16,8	89	109
B-n50-k8	50	1312	1448,1	10,4	64	140
B-n52-k7	52	747	779,1	4,3	45	78
B-n56-k7	56	707	814,6	15,2	48	140
B-n57-k9	57	1598	1743,0	9,1	106	281
B-n63-k10	63	1496	1783,2	19,2	175	468
B-n64-k9	64	861	1076,6	25,0	123	359
B-n66-k9	66	1316	1710,6	30,0	137	390
B-n67-k10	67	1032	1328,3	28,7	140	452
B-n68-k9	68	1272	1536,4	20,8	153	468
B-n78-k10	78	1221	1349,6	10,5	223	842
			<b>Média</b>	<b>16,6</b>	<b>74</b>	<b>282</b>

Fonte: Elaborado pelo autor

Para as instâncias da classe E, foram obtidos os resultados constantes da Tabela 4 abaixo:

Tabela 4 - Resultados para as instâncias da classe E

<b>Instância</b>	<b>Número de Pontos</b>	<b>Solução Ótima</b>	<b>Solução Obtida</b>	<b>Desvio da Solução (%)</b>	<b>Tempo com Parallel (ms)</b>	<b>Tempo sem Parallel (ms)</b>
E-n22-k4	22	375	473,5	26,3	4	7
E-n23-k3	23	569	650,0	14,2	5	11
E-n30-k3	30	534	552,8	3,5	5	11
E-n33-k4	33	835	948,0	13,5	10	14
E-n51-k5	51	521	640,8	23,0	34	47
E-n76-k10	76	830	979,9	18,1	278	905
E-n76-k14	76	1021	1271,1	24,5	651	1841
E-n76-k7	76	682	787,3	15,4	111	328
E-n76-k8	76	735	846,4	15,2	135	515
E-n101-k8	101	817	991,5	21,4	321	827
E-n101-k14	101	1071	1312,8	22,6	1040	3151
			<b>Média</b>	<b>18,0</b>	<b>236</b>	<b>696</b>

Fonte: Elaborado pelo autor

Para a única instância euclidiana da classe G, foi obtido o resultado constante da Tabela 5 a seguir:

Tabela 5 - Resultado para a instância da classe G

<b>Instância</b>	<b>Número de Pontos</b>	<b>Solução Ótima</b>	<b>Solução Obtida</b>	<b>Desvio da Solução (%)</b>	<b>Tempo com Parallel (ms)</b>	<b>Tempo sem Parallel (ms)</b>
G-n262-k25	262	6119	6745,8	10,2	42.501	149.136
<b>Média</b>				<b>10,2</b>	<b>42.501</b>	<b>149.136</b>

Fonte: Elaborado pelo autor

Para as instâncias da classe M, foram obtidos os resultados constantes da Tabela 6 abaixo:

Tabela 6 - Resultados para as instâncias da classe M

<b>Instância</b>	<b>Número de Pontos</b>	<b>Solução Ótima</b>	<b>Solução Obtida</b>	<b>Desvio da Solução (%)</b>	<b>Tempo com Parallel (ms)</b>	<b>Tempo sem Parallel (ms)</b>
M-n101-k10	101	820	841,6	2,6	474	1678
M-n121-k7	121	1034	1164,8	12,6	327	1154
M-n151-k12	151	1053	1273,5	20,9	1713	7348
M-n200-k17	200	1373	1557,8	13,5	9092	34928
<b>Média</b>				<b>12,4</b>	<b>2.901</b>	<b>11.277</b>

Fonte: Elaborado pelo autor

Para as instâncias da classe P, foram obtidos os resultados constantes da Tabela 7 a seguir:

Tabela 7 - Resultados para as instâncias da classe P

<b>Instância</b>	<b>Número de Pontos</b>	<b>Solução Ótima</b>	<b>Solução Obtida</b>	<b>Desvio da Solução (%)</b>	<b>Tempo com Parallel (ms)</b>	<b>Tempo sem Parallel (ms)</b>
P-n16-k8	16	450	477,2	6,0	0	0
P-n20-k2	20	216	227,0	5,1	0	0
P-n21-k2	21	211	234,5	11,1	0	0
P-n22-k2	22	216	242,0	12,0	0	0
P-n40-k5	40	458	529,6	15,6	20	31
P-n45-k5	45	510	561,1	10,0	20	31
P-n50-k10	50	696	819,4	17,7	60	265
P-n50-k7	50	554	633,1	14,3	40	142
P-n51-k10	51	741	838,8	13,2	100	343
P-n55-k10	55	694	815,2	17,5	100	437
P-n55-k7	55	568	641,8	13,0	40	156
P-n55-k8	55	588	711,6	21,0	120	281
P-n60-k10	60	744	885,8	19,1	150	530
P-n60-k15	60	968	1147,8	18,6	305	1186
P-n65-k10	65	792	930,7	17,5	191	608
P-n70-k10	70	827	980,8	18,6	180	764
P-n76-k4	76	593	735,7	24,1	50	78
P-n76-k5	76	627	760,1	21,2	70	187
P-n101-k4	101	681	804,4	18,1	80	203
<b>Média</b>				<b>15,5</b>	<b>80</b>	<b>276</b>

Fonte: Elaborado pelo autor

Para a única instância euclidiana da classe V, foi obtido o resultado constante da Tabela 8 a seguir:

Tabela 8 - Resultados para a instância da classe V

Instância	Número de Pontos	Solução Ótima	Solução Obtida	Desvio da Solução (%)	Tempo com <i>Parallel</i> (ms)	Tempo sem <i>Parallel</i> (ms)
att-n48-k4	48	40002	45881,0	14,7	20	31
<b>Média</b>				<b>14,7</b>	<b>20</b>	<b>31</b>

Fonte: Elaborado pelo autor

A Tabela 9 apresenta a síntese dos desvios globais obtidos em relação às soluções ótimas da instâncias, determinando um desvio médio geral de 15,0% para o presente trabalho.

Tabela 9 - Desvios médios obtidos na resolução do PRVC

Classe da Instância	Quantidade de Instâncias Analisadas	Desvio Médio em Relação à Solução Ótima (%)	Tempo médio de Processamento com <i>Parallelismo</i> (ms)	Tempo médio de Processamento sem <i>Parallelismo</i> (ms)
A	27	17,3	75	180
B	20	16,6	74	282
E	11	18,0	236	696
G	1	10,2	42.501	149.136
M	4	12,4	2.901	11.277
P	19	15,5	80	276
V	1	14,7	20	31
<b>Média</b>		<b>15,0</b>	<b>6.555</b>	<b>23.125</b>

Fonte: Elaborado pelo autor

Excetuando-se o tempo computacional gasto na instância G-n262-k25, todos os demais tempos obtidos foram inferiores a 03 segundos. Os 42,5 segundos gastos na instância supracitada justificam-se pela grande quantidade de clientes (262) e de veículos (25) presentes no problema resolvido.

Tais tempos computacionais corroboram a decisão tomada de empregar técnicas de processamento paralelo na implementação computacional da heurística utilizada.

Quanto ao desvio médio geral obtido, o valor de 15,0% representa uma média geral relevante para a heurística desenvolvida, principalmente em função da grande quantidade e variedade de instâncias testadas no estudo. Em várias instâncias foram obtidos desvios inferiores a 10% como, por exemplo, os apontados pelas instâncias M-n101-k10 (2,6%), B-n31-k5 (3,3%) e E-n30-k3 (3,5%).

A Figura 15, a seguir, apresenta a solução gráfica obtida para a instância M-n101-k10:



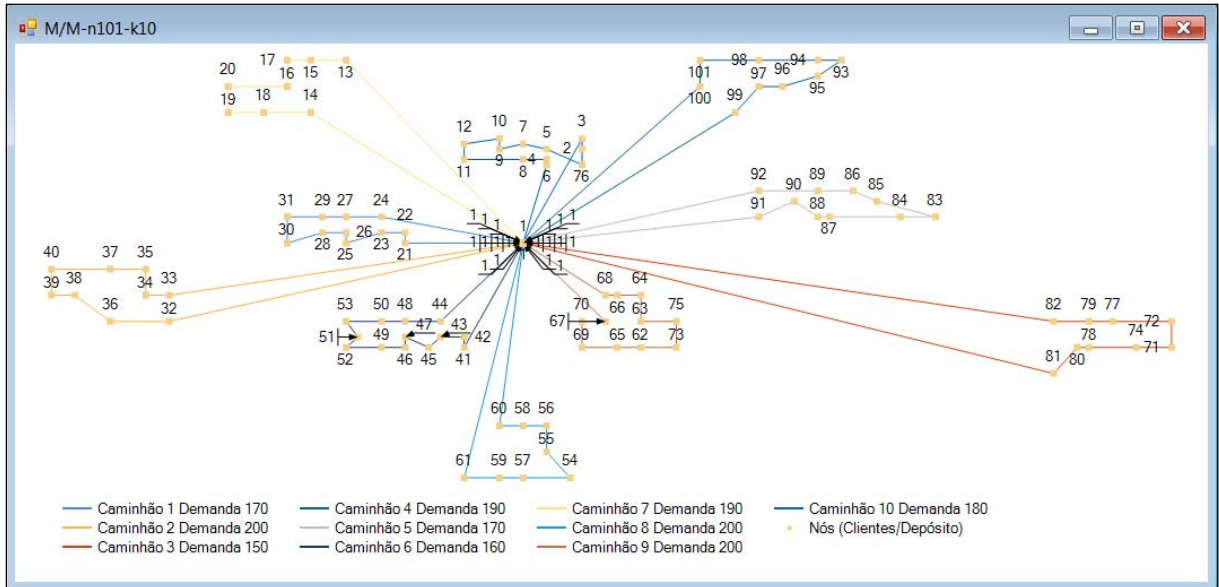


Figura 15 - Solução gráfica para a instância M-n101-k10

Fonte: Elaborado pelo autor

Para a instância B-n31-k5, foi obtida pelo *software* a solução gráfica exposta na Figura 16 abaixo:

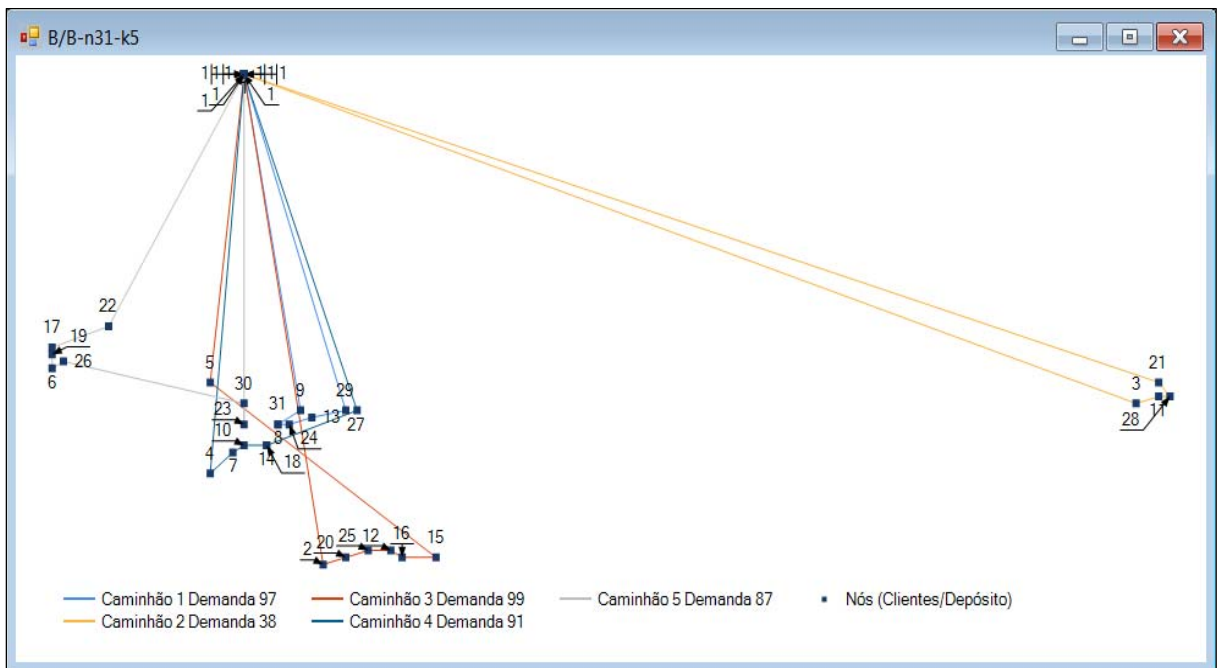


Figura 16 - Solução gráfica para a instância B-n31-k5

Fonte: Elaborado pelo autor

Para a instância E-n30-k3, o *software* obteve a solução gráfica apresentada na figura 17 a seguir:

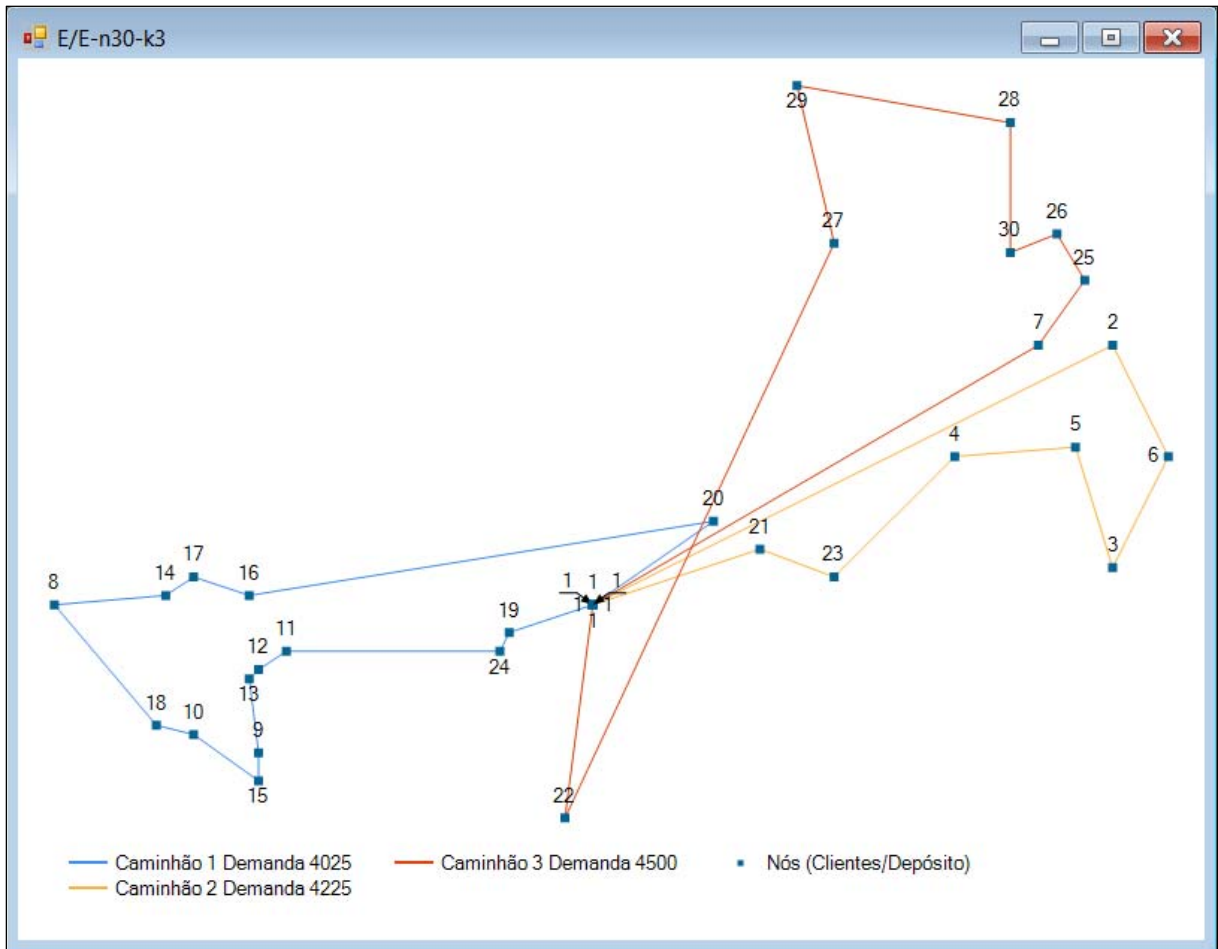


Figura 17 - Solução gráfica para a instância E-n30-k3  
 Fonte: Elaborado pelo autor

Os índices e os gráficos supracitados demonstram plenamente a qualidade da estratégia utilizada, a robustez do *software* desenvolvido e a boa precisão da heurística proposta neste trabalho.

## 5 CONCLUSÕES

O PRVC é um dos mais estudados e analisados problemas de otimização em redes, principalmente pela sua grande aplicabilidade nas áreas de Logística e *Supply Chain Management*. Porém, mesmo possuindo uma formulação exata relativamente simples, sua resolução analítica é de alta complexidade.

Dessa maneira, torna-se inviável o emprego da sua resolução analítica em problemas onde haja um grande número de clientes a serem atendidos e veículos a serem roteirizados. Tal contexto propicia a utilização de heurísticas como instrumentos fundamentais de resolução do PRVC, conforme demonstrado pelos resultados obtidos neste trabalho.

O presente estudo apresentou uma nova heurística baseada na estratégia “*Cluster First Route Second*” para resolver o PRVC. A estratégia utilizada diferencia-se por duas características básicas: a determinação de grupos de clientes a serem atendidos em função da capacidade do veículo e a roteirização do veículo dentro de cada grupo em função do algoritmo VMP.

Outro diferencial deste trabalho foi o emprego de técnicas de processamento paralelo (*Parallel Computing*) na implantação computacional da heurística. Essa implantação resultou em um tempo médio de processamento inferior a 7 segundos, evidenciando a ampla vantagem de sua utilização em problemas com alta complexidade de cálculos e dados.

No intuito de validar a precisão da heurística, foi determinado o desvio médio geral de 15% em relação aos resultados ótimos das instâncias testadas ao longo o estudo. Esse índice de precisão possibilita a aplicabilidade adequada da heurística proposta em problemas práticos das empresas, sem requerer das mesmas um tempo de processamento computacional elevado.

Por fim, como sugestão para futuros trabalhos, sugere-se o desenvolvimento de estratégias que visem aperfeiçoar a geração da solução inicial da heurística (problema das *p-medianas*). Há que se buscar esse aperfeiçoamento através da mudança do algoritmo inicial da heurística por métodos mais avançados como, por exemplo, simulações estocásticas ou algoritmos genéticos.

## REFERÊNCIAS

ACHUTHAN, N. R.; CACCETTA, L.; HILL, S. P. **An Improved Branch-and-Cut Algorithm for the Capacitated Vehicle Routing Problem**. *Transportation Science*, v. 37, n. 2, p. 153-169, Maio 2003.

AUGERAT, P.; BELENGUER, J.; BENAVENT, E.; CORBERÁN, A.; NADDEF, D.; RINALDI, G. **Computational Results with a Branch and Cut Code for the Capacitated Vehicle Routing Problem**. *Technical Report 949-M*, Université Joseph Fourier, Grenoble, France, 1995.

CAMPOS, D. S., **Integração de problemas de carregamento e roteamento de veículos, com janela de tempo e frota heterogênea**. 2008. 121 f. Tese (Doutorado em Engenharia de Produção) Universidade de São Paulo, São Paulo.

COSTA, D. M. B. **Aplicação de Algumas técnicas da Pesquisa Operacional na Otimização de Serviços Postais**. Dissertação (Mestrado em Métodos Numéricos em Engenharia), Universidade Federal do Paraná, Curitiba, 1997.

CAMPOS, V.; MOTA, E. **Heuristic Procedures for the Capacitated Vehicle Routing Problem**. *Computational Optimization and Applications*, v. 16, n. 3, p. 265-277, 2000.

CHRISTOFIDES, N.; MINGOZZI, A.; TOTH, P. **Exact algorithms for the vehicle routing problem, based on spanning tree and shortest path relaxations**. *Mathematical Programming*, v. 20, n. 1, p. 255-282, 1981.

COSTA, Deise Maria B. **Aplicação de algumas técnicas da pesquisa operacional na otimização de serviços postais**. Universidade Federal do Paraná. Curitiba. 1997.

COVER, T.; HART, P. **Nearest neighbor pattern classification**. IEEE Transactions on Information Theory, v. 13, n. 1, p. 21-27, 1967.

DANTZIG, G.; RAMSER, R. **The truck dispatching problem**. Management Science, v. 6, p. 80-91, 1959.

FISHER, M.; JAIKUMAR, R. **A generalized assignment heuristic for vehicle routing**. Networks, v. 11, p. 109–124, 1981.

FUKASAWA, Ricardo *et al.* **Robust Branch-and-Cut-and-Price for the Robust Branch-and-Cut-and-Price for the**. Mathematical Programming , v. 106, n. 3, p. 491-511, 2000.

GILLETT, Billy E.; JOHNSON, Jerry G. **Multi-terminal vehicle-dispatch algorithm**. Omega, v. 4, p. 711–717, 1976.

GUIMARÃES, Thiago A.; SCARPIN, Cassius T.; STEINER, Maria Teresinha A. **Uma nova abordagem heurística para a resolução do problema do roteamento de veículos capacitados**. Revista Gestão Industrial, v. 7, n. 3, p. 103-119, 2011.

HAKIMI, S. L. **Optimum distribution of switching centers in a communication network and some related graph theoretic problems**. Operations Research, v. 13, n. 3, p. 462-475, 1965.

LAPORTE, G.; GENDREAU, M.; POTVYN, J.; SEMET, F.. **Classical and modern heuristics for the vehicle routing problem**. International Transactions in Operational Research, v. 7, n. 4-5, p. 285-300, 1999.

LEIJEN , Daan; HALL, Judd. **Otimize o código gerenciado para máquinas de vários núcleos**. MSDN Magazine , 2007.

LIN, S.; KERNIGHAN, B. W. **An Effective Heuristic Algorithm for the Traveling-Salesman Problem**. *Operations Research*, v. 21, n. 2, p. 498-516, 1973.

LIN, Shih-Wei *et al.* **Applying hybrid meta-heuristics for capacitated vehicle routing problem**. *Expert Systems with Applications*, v. 36, n. 2, p. 1505-1512, 2009.

SARIKLIS, D.; POWELL, S. **A heuristic method for the open vehicle routing problem**. *Journal of the Operational*, v. 51, p. 564–573, 2000.

SLACK, Nigel.; CHAMBERS, Stuart.; JOHNSTON, Robert. **Administração da Produção**. 3. ed. São Paulo: Atlas, 2009.

STEINER, M.; ZAMBONI, L.; COSTA, D.; CARNIERI, C.; SILVA, A. **O Problema de Roteamento no Transporte Escolar**, *Revista Pesquisa Operacional*, v. 20, n. 01, p. 83-99, 2000.

TEITZ, M. B.; BART, P. **Heuristic Methods for Estimating the Generalized Vertex Median of a Weighted**. *Operations Research*, v. 16, p. 955-961, 1968.

TOUB, Stephen. **O passado, o presente e o futuro da paralelização de aplicativos.NET**. *MSDN Magazine*, 2011.

ZHISHUO, Liu; YUETING, Cai. **Sweep based multiple ant colonies algorithm for capacitated vehicle routing problem**. *IEEE International Conference on e-Business Engineering*, 2005. ICEBE 2005, Beijing, p. 387-394, 2005.

## APÊNDICES

### APÊNDICE A – ROTAS ENCONTRADAS PARA OS VEÍCULOS

Obs.: A numeração de identificação dos pontos (clientes) permanece como indicado no corpo das instâncias, onde o depósito é representado sempre como o ponto 1.

<b>Rotas Obtidas para os Veículos das 83 Instâncias Clássica Resolvidas</b>			
<b>Classe</b>	<b>Instância</b>	<b>Caminhão</b>	<b>Rota</b>
A	A-n32-k5	1	1, 21, 28, 25, 15, 7, 1
A	A-n32-k5	2	1, 14, 22, 32, 20, 18, 1
A	A-n32-k5	3	1, 4, 3, 24, 29, 9, 19, 23, 12, 5, 1
A	A-n32-k5	4	1, 6, 30, 26, 11, 16, 10, 1
A	A-n32-k5	5	1, 31, 27, 17, 13, 2, 8, 1
A	A-n33-k5	1	1, 7, 25, 21, 33, 14, 9, 8, 27, 1
A	A-n33-k5	2	1, 5, 17, 4, 10, 18, 1
A	A-n33-k5	3	1, 12, 19, 32, 20, 15, 22, 2, 30, 1
A	A-n33-k5	4	1, 29, 24, 16, 23, 3, 1
A	A-n33-k5	5	1, 13, 11, 31, 26, 28, 6, 1
A	A-n33-k6	1	1, 22, 18, 12, 30, 1
A	A-n33-k6	2	1, 29, 15, 11, 26, 17, 31, 28, 32, 1
A	A-n33-k6	3	1, 2, 19, 7, 8, 14, 6, 1
A	A-n33-k6	4	1, 33, 13, 1
A	A-n33-k6	5	1, 25, 24, 27, 23, 5, 9, 1
A	A-n33-k6	6	1, 20, 21, 10, 16, 3, 4, 1
A	A-n34-k5	1	1, 19, 3, 10, 13, 4, 23, 17, 34, 1
A	A-n34-k5	2	1, 18, 20, 26, 32, 29, 33, 1
A	A-n34-k5	3	1, 21, 5, 27, 15, 14, 22, 1
A	A-n34-k5	4	1, 6, 31, 25, 30, 24, 1
A	A-n34-k5	5	1, 11, 8, 7, 16, 9, 28, 2, 12, 1
A	A-n36-k5	1	1, 29, 13, 7, 4, 32, 20, 5, 1
A	A-n36-k5	2	1, 10, 15, 35, 24, 3, 36, 9, 1
A	A-n36-k5	3	1, 17, 2, 12, 25, 28, 26, 16, 1
A	A-n36-k5	4	1, 22, 23, 33, 19, 34, 30, 31, 18, 14, 1
A	A-n36-k5	5	1, 11, 8, 27, 21, 6, 1
A	A-n37-k5	1	1, 16, 22, 8, 17, 2, 4, 37, 1
A	A-n37-k5	2	1, 35, 31, 26, 36, 19, 27, 32, 29, 33, 30, 1
A	A-n37-k5	3	1, 23, 14, 5, 34, 6, 7, 1
A	A-n37-k5	4	1, 24, 25, 10, 9, 28, 12, 1
A	A-n37-k5	5	1, 13, 18, 15, 21, 20, 3, 11, 1
A	A-n37-k6	1	1, 19, 5, 18, 35, 32, 20, 21, 7, 1
A	A-n37-k6	2	1, 28, 36, 26, 33, 1
A	A-n37-k6	3	1, 16, 31, 11, 12, 13, 1
A	A-n37-k6	4	1, 25, 17, 8, 15, 37, 30, 1
A	A-n37-k6	5	1, 22, 23, 24, 29, 3, 10, 34, 1
A	A-n37-k6	6	1, 14, 27, 2, 4, 6, 9, 1
A	A-n38-k5	1	1, 8, 6, 23, 28, 12, 38, 1
A	A-n38-k5	2	1, 33, 19, 20, 7, 35, 30, 11, 22, 13, 1
A	A-n38-k5	3	1, 25, 15, 3, 10, 18, 37, 14, 16, 1
A	A-n38-k5	4	1, 21, 29, 32, 26, 17, 5, 2, 4, 1

A	A-n38-k5	5	1, 31, 27, 24, 36, 34, 9, 1
A	A-n39-k5	1	1, 18, 27, 35, 25, 3, 36, 38, 1
A	A-n39-k5	2	1, 15, 5, 19, 10, 13, 1
A	A-n39-k5	3	1, 31, 14, 29, 7, 37, 28, 1
A	A-n39-k5	4	1, 22, 9, 12, 39, 16, 30, 21, 24, 2, 32, 34, 20, 26, 1
A	A-n39-k5	5	1, 11, 33, 6, 17, 8, 4, 23, 1
A	A-n39-k6	1	1, 12, 7, 2, 37, 18, 24, 22, 1
A	A-n39-k6	2	1, 3, 9, 8, 5, 20, 34, 26, 1
A	A-n39-k6	3	1, 25, 4, 30, 29, 10, 6, 27, 1
A	A-n39-k6	4	1, 21, 33, 19, 23, 35, 28, 11, 17, 1
A	A-n39-k6	5	1, 16, 14, 31, 1
A	A-n39-k6	6	1, 13, 39, 38, 32, 15, 36, 1
A	A-n44-k6	1	1, 32, 5, 35, 18, 4, 13, 40, 1
A	A-n44-k6	2	1, 34, 27, 11, 12, 38, 43, 33, 22, 1
A	A-n44-k6	3	1, 3, 42, 15, 23, 37, 10, 39, 14, 1
A	A-n44-k6	4	1, 9, 16, 20, 25, 6, 28, 29, 8, 1
A	A-n44-k6	5	1, 30, 44, 2, 36, 19, 21, 17, 1
A	A-n44-k6	6	1, 7, 26, 41, 24, 31, 1
A	A-n45-k6	1	1, 7, 24, 26, 4, 11, 9, 42, 34, 22, 6, 1
A	A-n45-k6	2	1, 10, 27, 30, 44, 18, 19, 28, 1
A	A-n45-k6	3	1, 3, 13, 40, 37, 43, 5, 17, 23, 1
A	A-n45-k6	4	1, 25, 38, 35, 12, 41, 20, 31, 1
A	A-n45-k6	5	1, 29, 8, 33, 14, 21, 16, 1
A	A-n45-k6	6	1, 45, 2, 15, 36, 32, 39, 1
A	A-n45-k7	1	1, 25, 15, 45, 34, 28, 1
A	A-n45-k7	2	1, 11, 23, 6, 31, 38, 1
A	A-n45-k7	3	1, 10, 42, 44, 12, 4, 29, 1
A	A-n45-k7	4	1, 13, 33, 14, 19, 8, 9, 39, 17, 21, 41, 1
A	A-n45-k7	5	1, 32, 20, 37, 30, 1
A	A-n45-k7	6	1, 18, 24, 26, 16, 43, 2, 1
A	A-n45-k7	7	1, 36, 40, 35, 27, 22, 7, 5, 3, 1
A	A-n46-k7	1	1, 31, 3, 38, 11, 1
A	A-n46-k7	2	1, 13, 36, 26, 45, 19, 1
A	A-n46-k7	3	1, 12, 9, 34, 16, 25, 17, 43, 1
A	A-n46-k7	4	1, 24, 10, 29, 22, 18, 46, 1
A	A-n46-k7	5	1, 37, 6, 33, 4, 23, 8, 40, 20, 1
A	A-n46-k7	6	1, 42, 21, 28, 2, 41, 35, 32, 30, 1
A	A-n46-k7	7	1, 39, 15, 7, 44, 27, 14, 5, 1
A	A-n48-k7	1	1, 42, 48, 17, 11, 3, 1
A	A-n48-k7	2	1, 7, 23, 39, 26, 20, 38, 4, 1
A	A-n48-k7	3	1, 34, 31, 22, 30, 25, 47, 14, 5, 12, 1
A	A-n48-k7	4	1, 35, 28, 16, 9, 40, 27, 21, 43, 1
A	A-n48-k7	5	1, 33, 37, 41, 8, 46, 10, 29, 1
A	A-n48-k7	6	1, 15, 18, 19, 45, 36, 1
A	A-n48-k7	7	1, 24, 44, 32, 2, 6, 13, 1
A	A-n53-k7	1	1, 49, 43, 44, 24, 20, 33, 17, 1
A	A-n53-k7	2	1, 34, 7, 21, 32, 31, 1
A	A-n53-k7	3	1, 2, 5, 29, 23, 48, 10, 25, 42, 18, 8, 46, 16, 1
A	A-n53-k7	4	1, 39, 19, 41, 27, 11, 30, 50, 37, 3, 51, 1
A	A-n53-k7	5	1, 26, 40, 4, 6, 15, 13, 38, 1
A	A-n53-k7	6	1, 22, 35, 14, 53, 12, 1
A	A-n53-k7	7	1, 52, 28, 36, 9, 47, 45, 1



A	A-n54-k7	1	1, 31, 53, 17, 35, 15, 33, 7, 36, 1
A	A-n54-k7	2	1, 14, 54, 4, 23, 20, 1
A	A-n54-k7	3	1, 28, 3, 13, 38, 9, 41, 49, 1
A	A-n54-k7	4	1, 12, 39, 45, 10, 34, 22, 1
A	A-n54-k7	5	1, 30, 11, 37, 2, 18, 46, 27, 32, 1
A	A-n54-k7	6	1, 5, 42, 47, 43, 25, 52, 48, 26, 1
A	A-n54-k7	7	1, 44, 24, 21, 29, 19, 6, 8, 51, 40, 50, 16, 1
A	A-n55-k9	1	1, 8, 6, 54, 48, 39, 17, 41, 3, 1
A	A-n55-k9	2	1, 35, 12, 16, 52, 1
A	A-n55-k9	3	1, 19, 49, 51, 24, 53, 25, 45, 1
A	A-n55-k9	4	1, 31, 23, 20, 28, 29, 14, 55, 1
A	A-n55-k9	5	1, 9, 30, 42, 26, 5, 43, 38, 1
A	A-n55-k9	6	1, 22, 13, 33, 11, 34, 32, 1
A	A-n55-k9	7	1, 7, 2, 46, 44, 1
A	A-n55-k9	8	1, 37, 4, 15, 27, 21, 47, 18, 1
A	A-n55-k9	9	1, 36, 10, 50, 40, 1
A	A-n60-k9	1	1, 11, 55, 6, 46, 43, 1
A	A-n60-k9	2	1, 42, 19, 34, 15, 17, 21, 26, 25, 1
A	A-n60-k9	3	1, 44, 57, 13, 33, 10, 52, 1
A	A-n60-k9	4	1, 29, 50, 45, 49, 23, 1
A	A-n60-k9	5	1, 53, 39, 60, 8, 18, 1
A	A-n60-k9	6	1, 47, 41, 12, 32, 31, 54, 22, 5, 4, 1
A	A-n60-k9	7	1, 48, 35, 24, 59, 3, 2, 37, 1
A	A-n60-k9	8	1, 20, 9, 14, 30, 38, 58, 28, 1
A	A-n60-k9	9	1, 36, 56, 16, 27, 40, 51, 7, 1
A	A-n61-k9	1	1, 26, 21, 7, 42, 9, 54, 38, 1
A	A-n61-k9	2	1, 15, 56, 24, 8, 16, 1
A	A-n61-k9	3	1, 51, 34, 31, 33, 44, 27, 5, 58, 1
A	A-n61-k9	4	1, 4, 39, 11, 60, 1
A	A-n61-k9	5	1, 40, 14, 23, 20, 25, 59, 1
A	A-n61-k9	6	1, 43, 50, 3, 55, 47, 46, 1
A	A-n61-k9	7	1, 18, 29, 45, 61, 12, 32, 53, 1
A	A-n61-k9	8	1, 41, 6, 30, 37, 22, 28, 57, 2, 1
A	A-n61-k9	9	1, 10, 52, 48, 36, 19, 49, 17, 35, 13, 1
A	A-n62-k8	1	1, 44, 30, 62, 41, 7, 33, 58, 1
A	A-n62-k8	2	1, 53, 37, 17, 38, 23, 45, 3, 1
A	A-n62-k8	3	1, 60, 56, 27, 34, 59, 18, 51, 1
A	A-n62-k8	4	1, 61, 6, 19, 31, 2, 10, 55, 43, 24, 1
A	A-n62-k8	5	1, 46, 48, 25, 4, 40, 28, 1
A	A-n62-k8	6	1, 54, 13, 49, 57, 15, 14, 39, 20, 1
A	A-n62-k8	7	1, 35, 32, 26, 12, 50, 36, 5, 22, 42, 47, 1
A	A-n62-k8	8	1, 29, 16, 21, 8, 52, 11, 9, 1
A	A-n63-k10	1	1, 60, 11, 6, 19, 50, 46, 40, 4, 43, 20, 1
A	A-n63-k10	2	1, 38, 9, 36, 28, 7, 1
A	A-n63-k10	3	1, 15, 2, 42, 3, 10, 13, 1
A	A-n63-k10	4	1, 25, 56, 34, 54, 1
A	A-n63-k10	5	1, 22, 55, 61, 47, 1
A	A-n63-k10	6	1, 29, 33, 58, 18, 16, 1
A	A-n63-k10	7	1, 49, 31, 37, 53, 44, 1
A	A-n63-k10	8	1, 21, 5, 39, 27, 14, 51, 63, 1
A	A-n63-k10	9	1, 32, 52, 45, 48, 35, 12, 23, 59, 8, 1
A	A-n63-k10	10	1, 41, 26, 30, 57, 62, 24, 17, 1

A	A-n63-k9	1	1, 20, 57, 8, 26, 21, 14, 61, 17, 11, 1
A	A-n63-k9	2	1, 23, 18, 36, 51, 52, 38, 7, 1
A	A-n63-k9	3	1, 13, 22, 56, 33, 59, 41, 15, 58, 1
A	A-n63-k9	4	1, 46, 28, 60, 9, 25, 63, 1
A	A-n63-k9	5	1, 55, 30, 31, 35, 3, 6, 1
A	A-n63-k9	6	1, 12, 34, 16, 45, 29, 1
A	A-n63-k9	7	1, 27, 62, 50, 37, 49, 40, 1
A	A-n63-k9	8	1, 54, 24, 10, 43, 42, 4, 1
A	A-n63-k9	9	1, 53, 48, 5, 19, 47, 44, 32, 39, 2, 1
A	A-n64-k9	1	1, 24, 38, 55, 59, 3, 64, 47, 9, 48, 1
A	A-n64-k9	2	1, 54, 63, 21, 10, 35, 36, 1
A	A-n64-k9	3	1, 43, 33, 62, 25, 61, 17, 1
A	A-n64-k9	4	1, 52, 20, 5, 6, 1
A	A-n64-k9	5	1, 39, 19, 23, 13, 12, 31, 1
A	A-n64-k9	6	1, 50, 56, 60, 32, 51, 40, 45, 27, 22, 8, 37, 41, 1
A	A-n64-k9	7	1, 11, 42, 57, 53, 14, 49, 34, 1
A	A-n64-k9	8	1, 7, 26, 30, 46, 16, 15, 1
A	A-n64-k9	9	1, 4, 58, 18, 2, 44, 29, 28, 1
A	A-n65-k9	1	1, 35, 10, 23, 16, 15, 4, 1
A	A-n65-k9	2	1, 45, 57, 60, 9, 11, 1
A	A-n65-k9	3	1, 13, 25, 41, 59, 21, 33, 62, 43, 1
A	A-n65-k9	4	1, 6, 54, 42, 51, 17, 3, 39, 1
A	A-n65-k9	5	1, 30, 56, 19, 22, 26, 53, 20, 2, 34, 1
A	A-n65-k9	6	1, 18, 44, 28, 50, 63, 14, 1
A	A-n65-k9	7	1, 29, 24, 58, 49, 12, 64, 8, 55, 1
A	A-n65-k9	8	1, 52, 40, 47, 65, 7, 27, 32, 48, 1
A	A-n65-k9	9	1, 46, 61, 31, 38, 36, 37, 5, 1
A	A-n69-k9	1	1, 29, 35, 15, 58, 43, 26, 64, 51, 1
A	A-n69-k9	2	1, 41, 39, 57, 40, 9, 60, 42, 1
A	A-n69-k9	3	1, 20, 25, 23, 30, 13, 59, 67, 69, 1
A	A-n69-k9	4	1, 8, 68, 22, 62, 65, 32, 3, 34, 50, 1
A	A-n69-k9	5	1, 44, 53, 27, 55, 12, 47, 6, 49, 1
A	A-n69-k9	6	1, 28, 66, 56, 61, 5, 48, 1
A	A-n69-k9	7	1, 21, 46, 36, 14, 45, 16, 4, 52, 7, 1
A	A-n69-k9	8	1, 19, 54, 10, 31, 24, 17, 1
A	A-n69-k9	9	1, 38, 18, 33, 11, 37, 2, 63, 1
A	A-n80-k10	1	1, 63, 24, 13, 45, 6, 31, 25, 12, 1
A	A-n80-k10	2	1, 50, 74, 37, 43, 52, 54, 67, 68, 71, 1
A	A-n80-k10	3	1, 30, 60, 47, 26, 42, 65, 34, 16, 48, 57, 70, 66, 1
A	A-n80-k10	4	1, 21, 76, 62, 69, 44, 17, 58, 20, 27, 36, 1
A	A-n80-k10	5	1, 41, 22, 11, 8, 2, 14, 1
A	A-n80-k10	6	1, 59, 77, 73, 51, 46, 23, 33, 5, 1
A	A-n80-k10	7	1, 39, 55, 10, 56, 1
A	A-n80-k10	8	1, 72, 15, 53, 29, 49, 19, 80, 35, 1
A	A-n80-k10	9	1, 78, 4, 75, 40, 61, 32, 18, 1
A	A-n80-k10	10	1, 64, 7, 3, 38, 9, 79, 28, 1
B	B-n31-k5	1	1, 9, 31, 8, 24, 13, 29, 1
B	B-n31-k5	2	1, 3, 11, 28, 21, 1
B	B-n31-k5	3	1, 5, 15, 16, 12, 25, 20, 2, 1
B	B-n31-k5	4	1, 27, 14, 18, 10, 7, 4, 1
B	B-n31-k5	5	1, 22, 17, 19, 6, 26, 30, 23, 1
B	B-n34-k5	1	1, 12, 31, 8, 22, 30, 1

B	B-n34-k5	2	1, 2, 20, 19, 25, 5, 3, 1
B	B-n34-k5	3	1, 17, 10, 15, 9, 21, 1
B	B-n34-k5	4	1, 28, 4, 11, 24, 34, 33, 23, 1
B	B-n34-k5	5	1, 6, 7, 16, 14, 29, 32, 18, 26, 13, 27, 1
B	B-n35-k5	1	1, 30, 14, 3, 20, 12, 31, 16, 1
B	B-n35-k5	2	1, 27, 19, 7, 34, 4, 29, 33, 1
B	B-n35-k5	3	1, 18, 13, 21, 8, 10, 2, 24, 17, 11, 1
B	B-n35-k5	4	1, 23, 6, 15, 35, 28, 25, 5, 1
B	B-n35-k5	5	1, 22, 26, 32, 9, 1
B	B-n38-k6	1	1, 6, 5, 34, 19, 24, 31, 1
B	B-n38-k6	2	1, 7, 16, 2, 11, 21, 1
B	B-n38-k6	3	1, 27, 17, 23, 10, 20, 32, 26, 1
B	B-n38-k6	4	1, 12, 33, 4, 29, 22, 15, 1
B	B-n38-k6	5	1, 14, 38, 3, 36, 1
B	B-n38-k6	6	1, 8, 28, 13, 30, 25, 35, 9, 18, 37, 1
B	B-n39-k5	1	1, 32, 28, 39, 2, 27, 26, 33, 1
B	B-n39-k5	2	1, 30, 15, 4, 6, 16, 12, 13, 1
B	B-n39-k5	3	1, 3, 34, 22, 25, 38, 24, 31, 17, 23, 1
B	B-n39-k5	4	1, 10, 8, 35, 20, 21, 11, 14, 1
B	B-n39-k5	5	1, 9, 7, 19, 29, 37, 5, 36, 18, 1
B	B-n41-k6	1	1, 11, 6, 2, 12, 35, 34, 7, 1
B	B-n41-k6	2	1, 5, 40, 39, 28, 20, 27, 33, 1
B	B-n41-k6	3	1, 25, 36, 9, 14, 32, 1
B	B-n41-k6	4	1, 31, 8, 37, 17, 41, 21, 23, 1
B	B-n41-k6	5	1, 26, 3, 10, 38, 22, 19, 4, 1
B	B-n41-k6	6	1, 30, 15, 29, 18, 24, 13, 16, 1
B	B-n43-k6	1	1, 18, 40, 35, 8, 11, 32, 13, 39, 16, 17, 43, 1
B	B-n43-k6	2	1, 5, 36, 42, 28, 30, 33, 26, 41, 21, 34, 6, 1
B	B-n43-k6	3	1, 4, 12, 31, 14, 1
B	B-n43-k6	4	1, 27, 9, 7, 25, 23, 1
B	B-n43-k6	5	1, 22, 19, 20, 15, 29, 38, 1
B	B-n43-k6	6	1, 10, 37, 3, 24, 2, 1
B	B-n44-k7	1	1, 5, 9, 31, 43, 23, 39, 1
B	B-n44-k7	2	1, 7, 22, 14, 35, 38, 42, 10, 1
B	B-n44-k7	3	1, 16, 34, 44, 32, 28, 36, 1
B	B-n44-k7	4	1, 12, 11, 24, 1
B	B-n44-k7	5	1, 3, 37, 15, 27, 25, 40, 1
B	B-n44-k7	6	1, 2, 13, 4, 29, 26, 21, 1
B	B-n44-k7	7	1, 8, 17, 19, 30, 33, 20, 41, 18, 6, 1
B	B-n45-k5	1	1, 9, 22, 5, 17, 30, 12, 26, 27, 20, 1
B	B-n45-k5	2	1, 38, 33, 16, 3, 14, 29, 41, 1
B	B-n45-k5	3	1, 6, 7, 44, 4, 32, 39, 21, 36, 25, 1
B	B-n45-k5	4	1, 43, 10, 45, 8, 35, 28, 15, 19, 1
B	B-n45-k5	5	1, 40, 18, 31, 24, 2, 37, 11, 42, 23, 13, 34, 1
B	B-n50-k7	1	1, 8, 31, 19, 15, 44, 11, 43, 4, 17, 28, 25, 1
B	B-n50-k7	2	1, 35, 50, 41, 16, 7, 18, 20, 26, 1
B	B-n50-k7	3	1, 29, 33, 3, 49, 45, 1
B	B-n50-k7	4	1, 30, 47, 38, 12, 21, 9, 1
B	B-n50-k7	5	1, 22, 46, 37, 6, 1
B	B-n50-k7	6	1, 5, 2, 14, 27, 40, 1
B	B-n50-k7	7	1, 48, 34, 23, 36, 24, 10, 39, 13, 32, 42, 1
B	B-n50-k8	1	1, 38, 28, 13, 34, 10, 50, 44, 37, 21, 31, 1

B	B-n50-k8	2	1, 12, 23, 39, 47, 24, 36, 11, 4, 1
B	B-n50-k8	3	1, 46, 6, 1
B	B-n50-k8	4	1, 7, 32, 40, 16, 1
B	B-n50-k8	5	1, 15, 17, 25, 5, 48, 27, 19, 1
B	B-n50-k8	6	1, 42, 22, 9, 43, 18, 1
B	B-n50-k8	7	1, 2, 33, 29, 30, 20, 1
B	B-n50-k8	8	1, 3, 35, 45, 49, 41, 8, 26, 14, 1
B	B-n52-k7	1	1, 34, 36, 8, 44, 3, 1
B	B-n52-k7	2	1, 27, 14, 47, 33, 35, 20, 16, 50, 18, 1
B	B-n52-k7	3	1, 45, 11, 30, 28, 9, 6, 52, 1
B	B-n52-k7	4	1, 43, 21, 19, 31, 2, 1
B	B-n52-k7	5	1, 4, 29, 12, 22, 15, 40, 25, 32, 1
B	B-n52-k7	6	1, 42, 38, 48, 7, 26, 37, 41, 1
B	B-n52-k7	7	1, 49, 39, 10, 17, 23, 5, 46, 24, 13, 51, 1
B	B-n56-k7	1	1, 20, 19, 47, 29, 52, 15, 18, 9, 11, 44, 1
B	B-n56-k7	2	1, 49, 37, 4, 14, 34, 50, 33, 1
B	B-n56-k7	3	1, 48, 22, 27, 40, 5, 38, 1
B	B-n56-k7	4	1, 41, 45, 24, 23, 51, 39, 13, 10, 1
B	B-n56-k7	5	1, 54, 7, 25, 42, 46, 43, 30, 16, 8, 1
B	B-n56-k7	6	1, 3, 31, 35, 55, 32, 21, 56, 12, 1
B	B-n56-k7	7	1, 28, 36, 17, 53, 2, 26, 6, 1
B	B-n57-k9	1	1, 9, 16, 26, 52, 38, 17, 1
B	B-n57-k9	2	1, 8, 43, 12, 15, 35, 45, 14, 1
B	B-n57-k9	3	1, 4, 47, 11, 18, 32, 56, 51, 1
B	B-n57-k9	4	1, 44, 55, 46, 53, 25, 1
B	B-n57-k9	5	1, 6, 48, 37, 22, 42, 30, 54, 5, 1
B	B-n57-k9	6	1, 10, 29, 19, 28, 2, 24, 1
B	B-n57-k9	7	1, 7, 57, 23, 33, 41, 1
B	B-n57-k9	8	1, 21, 3, 27, 40, 20, 1
B	B-n57-k9	9	1, 31, 50, 39, 13, 49, 36, 34, 1
B	B-n63-k10	1	1, 5, 50, 57, 60, 47, 9, 40, 1
B	B-n63-k10	2	1, 19, 51, 2, 35, 1
B	B-n63-k10	3	1, 11, 58, 39, 54, 24, 12, 18, 1
B	B-n63-k10	4	1, 53, 20, 52, 22, 34, 26, 1
B	B-n63-k10	5	1, 31, 7, 42, 15, 59, 55, 13, 25, 44, 1
B	B-n63-k10	6	1, 6, 56, 17, 41, 32, 1
B	B-n63-k10	7	1, 46, 21, 8, 37, 28, 1
B	B-n63-k10	8	1, 29, 49, 61, 14, 33, 30, 1
B	B-n63-k10	9	1, 38, 23, 43, 62, 63, 3, 27, 36, 1
B	B-n63-k10	10	1, 45, 4, 16, 48, 10, 1
B	B-n64-k9	1	1, 4, 10, 38, 20, 6, 52, 56, 1
B	B-n64-k9	2	1, 16, 25, 5, 15, 60, 36, 37, 49, 61, 23, 1
B	B-n64-k9	3	1, 13, 34, 27, 21, 43, 64, 3, 1
B	B-n64-k9	4	1, 18, 55, 9, 22, 28, 35, 39, 29, 1
B	B-n64-k9	5	1, 31, 32, 41, 17, 14, 57, 11, 1
B	B-n64-k9	6	1, 47, 63, 53, 50, 44, 1
B	B-n64-k9	7	1, 8, 42, 2, 24, 26, 58, 40, 1
B	B-n64-k9	8	1, 33, 19, 51, 62, 48, 30, 1
B	B-n64-k9	9	1, 12, 45, 46, 59, 54, 7, 1
B	B-n66-k9	1	1, 33, 59, 6, 22, 7, 1
B	B-n66-k9	2	1, 24, 35, 12, 8, 28, 43, 1
B	B-n66-k9	3	1, 58, 60, 51, 25, 13, 50, 1

B	B-n66-k9	4	1, 53, 56, 32, 20, 57, 23, 44, 14, 1
B	B-n66-k9	5	1, 52, 29, 27, 15, 40, 30, 4, 16, 61, 5, 1
B	B-n66-k9	6	1, 48, 65, 45, 47, 11, 37, 1
B	B-n66-k9	7	1, 34, 66, 41, 9, 10, 3, 1
B	B-n66-k9	8	1, 2, 42, 21, 54, 55, 18, 19, 46, 1
B	B-n66-k9	9	1, 39, 36, 31, 62, 17, 63, 26, 49, 38, 64, 1
B	B-n67-k10	1	1, 17, 42, 44, 28, 11, 39, 18, 47, 1
B	B-n67-k10	2	1, 63, 62, 31, 58, 4, 1
B	B-n67-k10	3	1, 29, 14, 13, 59, 16, 57, 1
B	B-n67-k10	4	1, 52, 8, 61, 1
B	B-n67-k10	5	1, 51, 12, 36, 50, 35, 7, 1
B	B-n67-k10	6	1, 5, 40, 54, 43, 45, 41, 65, 37, 1
B	B-n67-k10	7	1, 34, 48, 67, 27, 19, 32, 53, 1
B	B-n67-k10	8	1, 55, 30, 60, 25, 15, 21, 20, 2, 1
B	B-n67-k10	9	1, 3, 33, 66, 38, 24, 9, 1
B	B-n67-k10	10	1, 6, 64, 10, 49, 26, 22, 56, 23, 46, 1
B	B-n68-k9	1	1, 38, 9, 3, 23, 50, 58, 1
B	B-n68-k9	2	1, 66, 67, 44, 47, 18, 27, 1
B	B-n68-k9	3	1, 65, 61, 36, 48, 8, 14, 1
B	B-n68-k9	4	1, 64, 51, 41, 40, 24, 56, 39, 54, 1
B	B-n68-k9	5	1, 20, 26, 2, 57, 34, 16, 32, 28, 1
B	B-n68-k9	6	1, 12, 4, 30, 68, 33, 7, 37, 43, 5, 1
B	B-n68-k9	7	1, 31, 60, 17, 45, 35, 46, 13, 49, 52, 1
B	B-n68-k9	8	1, 6, 15, 63, 22, 10, 21, 53, 42, 62, 1
B	B-n68-k9	9	1, 55, 29, 19, 25, 11, 59, 1
B	B-n78-k10	1	1, 10, 20, 5, 34, 56, 36, 60, 37, 15, 42, 1
B	B-n78-k10	2	1, 2, 18, 57, 48, 66, 50, 64, 6, 31, 1
B	B-n78-k10	3	1, 3, 43, 61, 67, 17, 19, 26, 11, 69, 1
B	B-n78-k10	4	1, 9, 23, 72, 16, 13, 35, 52, 1
B	B-n78-k10	5	1, 30, 46, 47, 4, 55, 21, 1
B	B-n78-k10	6	1, 28, 76, 38, 8, 77, 33, 74, 32, 1
B	B-n78-k10	7	1, 58, 71, 63, 7, 1
B	B-n78-k10	8	1, 51, 62, 40, 14, 49, 75, 29, 1
B	B-n78-k10	9	1, 78, 24, 45, 27, 12, 65, 41, 54, 1
B	B-n78-k10	10	1, 73, 59, 39, 70, 68, 44, 22, 25, 53, 1
E	E-n101-k14	1	1, 3, 74, 16, 42, 23, 76, 75, 73, 22, 1
E	E-n101-k14	2	1, 59, 14, 96, 95, 7, 97, 38, 1
E	E-n101-k14	3	1, 98, 88, 43, 101, 92, 45, 15, 44, 57, 1
E	E-n101-k14	4	1, 77, 78, 4, 69, 81, 55, 25, 30, 1
E	E-n101-k14	5	1, 5, 56, 26, 40, 24, 68, 1
E	E-n101-k14	6	1, 51, 21, 72, 66, 67, 33, 1
E	E-n101-k14	7	1, 32, 89, 63, 11, 64, 91, 31, 71, 1
E	E-n101-k14	8	1, 54, 41, 27, 13, 29, 28, 70, 2, 1
E	E-n101-k14	9	1, 34, 52, 10, 82, 80, 79, 35, 36, 1
E	E-n101-k14	10	1, 12, 65, 50, 37, 1
E	E-n101-k14	11	1, 8, 83, 49, 20, 48, 47, 1
E	E-n101-k14	12	1, 90, 61, 6, 85, 18, 46, 9, 84, 19, 53, 1
E	E-n101-k14	13	1, 58, 62, 17, 87, 39, 1
E	E-n101-k14	14	1, 100, 60, 93, 99, 94, 86, 1
E	E-n101-k8	1	1, 74, 75, 73, 5, 26, 40, 68, 24, 57, 76, 23, 42, 1
E	E-n101-k8	2	1, 27, 13, 69, 81, 55, 56, 25, 30, 80, 4, 78, 77, 51, 1
E	E-n101-k8	3	1, 34, 82, 52, 21, 67, 72, 10, 79, 35, 36, 66, 1

E	E-n101-k8	4	1, 54, 28, 90, 7, 95, 96, 88, 58, 3, 22, 41, 59, 14, 1
E	E-n101-k8	5	1, 29, 70, 2, 71, 31, 11, 32, 89, 63, 91, 33, 64, 12, 65, 1
E	E-n101-k8	6	1, 53, 19, 84, 9, 83, 8, 49, 20, 50, 37, 48, 47, 46, 1
E	E-n101-k8	7	1, 98, 43, 16, 44, 15, 45, 17, 87, 39, 1
E	E-n101-k8	8	1, 97, 100, 60, 93, 38, 99, 94, 86, 101, 92, 62, 18, 85, 6, 61, 1
E	E-n22-k4	1	1, 11, 10, 8, 6, 3, 2, 1
E	E-n22-k4	2	1, 9, 4, 5, 12, 20, 1
E	E-n22-k4	3	1, 14, 17, 15, 13, 7, 1
E	E-n22-k4	4	1, 16, 19, 21, 18, 22, 1
E	E-n23-k3	1	1, 12, 7, 10, 6, 5, 9, 8, 22, 1
E	E-n23-k3	2	1, 13, 19, 20, 21, 23, 18, 15, 16, 17, 4, 3, 1
E	E-n23-k3	3	1, 14, 11, 2, 1
E	E-n30-k3	1	1, 19, 24, 11, 12, 13, 9, 15, 10, 18, 8, 14, 17, 16, 20, 1
E	E-n30-k3	2	1, 21, 23, 4, 5, 3, 6, 2, 1
E	E-n30-k3	3	1, 22, 27, 29, 28, 30, 26, 25, 7, 1
E	E-n33-k4	1	1, 17, 29, 28, 27, 26, 1
E	E-n33-k4	2	1, 5, 14, 16, 2, 32, 31, 30, 1
E	E-n33-k4	3	1, 15, 18, 20, 19, 22, 21, 23, 25, 24, 1
E	E-n33-k4	4	1, 4, 3, 13, 12, 33, 11, 10, 9, 8, 7, 6, 1
E	E-n51-k5	1	1, 47, 33, 2, 23, 9, 27, 32, 29, 4, 37, 36, 21, 30, 1
E	E-n51-k5	2	1, 28, 7, 19, 26, 15, 25, 24, 44, 1
E	E-n51-k5	3	1, 39, 10, 51, 17, 3, 22, 35, 31, 11, 50, 1
E	E-n51-k5	4	1, 6, 5, 43, 20, 41, 42, 14, 49, 8, 1
E	E-n51-k5	5	1, 13, 48, 18, 38, 45, 16, 46, 34, 40, 12, 1
E	E-n76-k10	1	1, 27, 13, 41, 10, 40, 73, 32, 1
E	E-n76-k10	2	1, 64, 2, 44, 43, 65, 42, 57, 24, 50, 1
E	E-n76-k10	3	1, 36, 12, 59, 39, 66, 67, 1
E	E-n76-k10	4	1, 68, 35, 47, 53, 28, 14, 9, 1
E	E-n76-k10	5	1, 55, 58, 16, 38, 21, 71, 61, 72, 22, 1
E	E-n76-k10	6	1, 46, 3, 75, 29, 63, 74, 23, 62, 1
E	E-n76-k10	7	1, 8, 54, 20, 15, 60, 11, 1
E	E-n76-k10	8	1, 5, 69, 7, 18, 52, 17, 34, 1
E	E-n76-k10	9	1, 4, 45, 33, 51, 26, 56, 19, 25, 1
E	E-n76-k10	10	1, 76, 31, 49, 48, 70, 37, 6, 30, 1
E	E-n76-k14	1	1, 4, 45, 33, 51, 26, 56, 1
E	E-n76-k14	2	1, 3, 7, 50, 25, 40, 1
E	E-n76-k14	3	1, 68, 20, 15, 60, 1
E	E-n76-k14	4	1, 44, 43, 65, 42, 57, 1
E	E-n76-k14	5	1, 36, 12, 66, 67, 1
E	E-n76-k14	6	1, 35, 47, 53, 9, 8, 1
E	E-n76-k14	7	1, 73, 59, 39, 11, 32, 1
E	E-n76-k14	8	1, 18, 13, 41, 10, 1
E	E-n76-k14	9	1, 46, 31, 49, 30, 28, 1
E	E-n76-k14	10	1, 52, 17, 64, 34, 74, 2, 24, 1
E	E-n76-k14	11	1, 63, 23, 62, 29, 75, 55, 1
E	E-n76-k14	12	1, 14, 6, 48, 22, 37, 70, 1
E	E-n76-k14	13	1, 58, 16, 38, 21, 71, 61, 72, 19, 1
E	E-n76-k14	14	1, 76, 69, 5, 27, 54, 1
E	E-n76-k7	1	1, 18, 52, 45, 4, 17, 50, 25, 19, 51, 33, 41, 1
E	E-n76-k7	2	1, 34, 74, 2, 23, 65, 43, 42, 44, 64, 24, 57, 1

E	E-n76-k7	3	1, 36, 54, 15, 12, 39, 66, 67, 60, 55, 1
E	E-n76-k7	4	1, 13, 73, 40, 10, 26, 56, 32, 11, 59, 1
E	E-n76-k7	5	1, 69, 7, 3, 31, 75, 63, 29, 62, 22, 49, 46, 1
E	E-n76-k7	6	1, 30, 6, 38, 48, 37, 70, 72, 61, 71, 21, 16, 58, 14, 20, 1
E	E-n76-k7	7	1, 76, 5, 68, 35, 28, 53, 47, 9, 8, 27, 1
E	E-n76-k8	1	1, 30, 6, 38, 48, 37, 70, 72, 61, 71, 21, 16, 58, 1
E	E-n76-k8	2	1, 13, 73, 40, 10, 26, 56, 32, 11, 59, 1
E	E-n76-k8	3	1, 31, 3, 63, 29, 75, 49, 22, 62, 1
E	E-n76-k8	4	1, 8, 36, 54, 12, 39, 66, 67, 60, 1
E	E-n76-k8	5	1, 34, 74, 2, 23, 65, 43, 42, 44, 64, 24, 57, 1
E	E-n76-k8	6	1, 76, 69, 7, 52, 18, 27, 68, 5, 46, 1
E	E-n76-k8	7	1, 41, 33, 45, 4, 17, 50, 25, 19, 51, 1
E	E-n76-k8	8	1, 35, 47, 53, 28, 14, 55, 9, 20, 15, 1
F	F-n135-k7	1	1, 65, 78, 73, 22, 66, 120, 116, 117, 118, 114, 130, 129, 128, 127, 113, 126, 112, 125, 124, 122, 123, 111, 82, 1
F	F-n135-k7	2	1, 74, 135, 33, 76, 24, 27, 60, 61, 2, 49, 35, 80, 134, 68, 71, 70, 69, 81, 34, 20, 131, 1
F	F-n135-k7	3	1, 133, 132, 121, 110, 1
F	F-n135-k7	4	1, 79, 64, 77, 48, 29, 17, 14, 11, 3, 6, 10, 9, 8, 12, 13, 16, 15, 89, 88, 90, 91, 28, 26, 25, 23, 92, 83, 21, 87, 85, 86, 84, 1
F	F-n135-k7	5	1, 59, 55, 56, 104, 57, 58, 30, 7, 43, 42, 45, 44, 41, 96, 38, 39, 40, 46, 95, 94, 106, 98, 97, 99, 101, 100, 37, 36, 102, 105, 103, 54, 52, 51, 1
F	F-n135-k7	6	1, 75, 115, 107, 108, 109, 1
F	F-n135-k7	7	1, 47, 119, 19, 18, 72, 67, 50, 63, 53, 62, 32, 31, 93, 5, 4, 1
F	F-n45-k4	1	1, 45, 44, 43, 10, 3, 19, 15, 13, 12, 14, 11, 36, 41, 1
F	F-n45-k4	2	1, 25, 16, 21, 20, 18, 17, 2, 38, 39, 4, 5, 6, 7, 28, 1
F	F-n45-k4	3	1, 37, 40, 42, 31, 35, 32, 33, 34, 30, 8, 1
F	F-n45-k4	4	1, 9, 26, 22, 27, 24, 23, 29, 1
F	F-n72-k4	1	1, 37, 12, 2, 16, 3, 18, 14, 13, 17, 1
F	F-n72-k4	2	1, 35, 55, 56, 42, 58, 57, 61, 62, 59, 60, 64, 63, 15, 36, 19, 20, 65, 66, 67, 68, 39, 69, 40, 53, 1
F	F-n72-k4	3	1, 32, 6, 4, 5, 10, 8, 9, 11, 7, 72, 70, 41, 71, 52, 50, 48, 49, 1
F	F-n72-k4	4	1, 34, 33, 21, 30, 24, 25, 26, 27, 31, 22, 23, 29, 28, 43, 45, 47, 44, 54, 46, 51, 38, 1
G	G-n262-k25	1	1, 113, 257, 189, 96, 29, 183, 35, 88, 155, 245, 1
G	G-n262-k25	2	1, 84, 184, 151, 37, 221, 207, 66, 229, 1
G	G-n262-k25	3	1, 235, 69, 143, 233, 100, 71, 77, 156, 232, 203, 140, 72, 32, 230, 2, 1
G	G-n262-k25	4	1, 214, 216, 117, 127, 105, 75, 142, 26, 200, 251, 209, 247, 8, 165, 141, 173, 1
G	G-n262-k25	5	1, 256, 125, 50, 13, 134, 6, 212, 91, 62, 131, 130, 222, 206, 1
G	G-n262-k25	6	1, 157, 51, 89, 152, 95, 150, 87, 248, 1
G	G-n262-k25	7	1, 126, 102, 204, 58, 215, 132, 239, 11, 12, 242, 236, 1
G	G-n262-k25	8	1, 187, 82, 5, 116, 146, 101, 210, 139, 253, 1
G	G-n262-k25	9	1, 45, 23, 121, 237, 24, 129, 144, 175, 49, 197, 112,

			258, 254, 1
G	G-n262-k25	10	1, 135, 179, 118, 241, 99, 108, 15, 46, 1
G	G-n262-k25	11	1, 10, 154, 249, 190, 107, 122, 90, 168, 14, 1
G	G-n262-k25	12	1, 17, 244, 7, 38, 166, 223, 86, 93, 1
G	G-n262-k25	13	1, 43, 198, 195, 81, 114, 104, 169, 48, 56, 176, 67, 1
G	G-n262-k25	14	1, 259, 79, 33, 202, 68, 133, 98, 52, 246, 148, 1
G	G-n262-k25	15	1, 196, 3, 171, 218, 59, 231, 27, 149, 115, 252, 1
G	G-n262-k25	16	1, 109, 36, 260, 240, 120, 219, 185, 193, 145, 92, 1
G	G-n262-k25	17	1, 226, 227, 18, 25, 41, 128, 160, 213, 1
G	G-n262-k25	18	1, 205, 199, 74, 97, 167, 208, 31, 19, 138, 55, 261, 28, 174, 54, 1
G	G-n262-k25	19	1, 16, 220, 53, 170, 182, 172, 60, 163, 76, 234, 1
G	G-n262-k25	20	1, 63, 123, 44, 224, 255, 57, 243, 39, 40, 103, 177, 137, 1
G	G-n262-k25	21	1, 70, 178, 238, 147, 85, 136, 4, 119, 124, 191, 9, 161, 1
G	G-n262-k25	22	1, 164, 225, 250, 158, 83, 65, 194, 1
G	G-n262-k25	23	1, 192, 262, 80, 20, 186, 73, 34, 22, 1
G	G-n262-k25	24	1, 201, 64, 153, 181, 217, 47, 180, 162, 94, 188, 78, 1
G	G-n262-k25	25	1, 106, 30, 228, 21, 110, 159, 61, 111, 42, 211, 1
M	M-n101-k10	1	1, 21, 22, 23, 25, 26, 28, 30, 31, 29, 27, 24, 1
M	M-n101-k10	2	1, 33, 34, 35, 37, 40, 39, 38, 36, 32, 1
M	M-n101-k10	3	1, 82, 79, 77, 72, 71, 74, 78, 80, 81, 1
M	M-n101-k10	4	1, 99, 97, 96, 95, 93, 94, 98, 101, 100, 1
M	M-n101-k10	5	1, 91, 90, 88, 87, 84, 83, 85, 86, 89, 92, 1
M	M-n101-k10	6	1, 44, 48, 50, 53, 51, 52, 49, 46, 47, 45, 43, 42, 41, 1
M	M-n101-k10	7	1, 14, 18, 19, 20, 16, 17, 15, 13, 1
M	M-n101-k10	8	1, 60, 58, 56, 55, 54, 57, 59, 61, 1
M	M-n101-k10	9	1, 68, 66, 64, 63, 75, 73, 62, 65, 69, 70, 67, 1
M	M-n101-k10	10	1, 6, 4, 8, 11, 12, 10, 9, 7, 5, 76, 2, 3, 1
M	M-n121-k7	1	1, 103, 68, 70, 71, 72, 75, 73, 76, 79, 78, 77, 74, 69, 80, 81, 1
M	M-n121-k7	2	1, 106, 22, 27, 29, 24, 21, 18, 17, 20, 26, 23, 25, 28, 34, 31, 32, 35, 37, 36, 33, 30, 1
M	M-n121-k7	3	1, 89, 83, 112, 87, 88, 90, 93, 107, 121, 120, 82, 113, 86, 85, 118, 114, 84, 109, 1
M	M-n121-k7	4	1, 96, 41, 44, 46, 49, 48, 43, 40, 39, 38, 42, 45, 47, 50, 51, 52, 1
M	M-n121-k7	5	1, 7, 8, 10, 11, 12, 9, 13, 14, 15, 16, 5, 4, 6, 3, 2, 1
M	M-n121-k7	6	1, 102, 100, 101, 117, 98, 95, 97, 94, 92, 91, 115, 19, 119, 110, 116, 111, 99, 104, 105, 108, 1
M	M-n121-k7	7	1, 53, 55, 58, 60, 66, 62, 63, 65, 59, 56, 54, 57, 61, 64, 67, 1
M	M-n151-k12	1	1, 71, 123, 31, 21, 66, 67, 129, 132, 33, 124, 1
M	M-n151-k12	2	1, 34, 82, 80, 130, 79, 35, 136, 36, 137, 72, 104, 10, 121, 52, 1
M	M-n151-k12	3	1, 54, 106, 41, 59, 14, 95, 7, 148, 90, 113, 147, 53, 1
M	M-n151-k12	4	1, 128, 32, 89, 149, 63, 11, 109, 91, 64, 127, 12, 108, 8, 107, 1
M	M-n151-k12	5	1, 13, 110, 81, 151, 69, 117, 78, 4, 122, 30, 25, 135, 55, 1
M	M-n151-k12	6	1, 131, 56, 26, 140, 40, 68, 24, 57, 5, 1
M	M-n151-k12	7	1, 28, 29, 112, 133, 70, 102, 2, 51, 103, 77, 139, 27,



			150, 1
M	M-n151-k12	8	1, 118, 96, 98, 93, 60, 97, 105, 100, 94, 99, 38, 101, 43, 145, 88, 1
M	M-n151-k12	9	1, 19, 84, 61, 119, 85, 115, 9, 126, 46, 18, 86, 1
M	M-n151-k12	10	1, 62, 17, 92, 15, 120, 45, 142, 114, 87, 141, 39, 44, 1
M	M-n151-k12	11	1, 6, 83, 49, 125, 48, 20, 65, 50, 144, 37, 47, 1
M	M-n151-k12	12	1, 138, 3, 116, 58, 146, 75, 73, 74, 22, 111, 76, 134, 23, 42, 16, 143, 1
M	M-n200-k17	1	1, 49, 125, 48, 169, 124, 20, 108, 50, 144, 37, 1
M	M-n200-k17	2	1, 51, 158, 34, 82, 186, 80, 130, 170, 79, 35, 165, 121, 1
M	M-n200-k17	3	1, 113, 14, 118, 96, 97, 95, 184, 7, 148, 90, 28, 1
M	M-n200-k17	4	1, 138, 88, 145, 3, 116, 179, 58, 173, 43, 143, 44, 16, 1
M	M-n200-k17	5	1, 157, 19, 167, 61, 119, 6, 62, 174, 18, 85, 84, 1
M	M-n200-k17	6	1, 22, 74, 172, 75, 73, 199, 111, 198, 76, 134, 23, 146, 42, 1
M	M-n200-k17	7	1, 98, 93, 152, 60, 100, 105, 94, 86, 92, 194, 101, 38, 99, 1
M	M-n200-k17	8	1, 177, 2, 71, 123, 31, 21, 189, 129, 161, 132, 33, 91, 182, 1
M	M-n200-k17	9	1, 183, 149, 63, 160, 11, 190, 109, 127, 64, 12, 176, 65, 1
M	M-n200-k17	10	1, 193, 192, 142, 17, 114, 87, 45, 120, 15, 141, 39, 1
M	M-n200-k17	11	1, 54, 106, 29, 155, 139, 27, 181, 41, 59, 153, 1
M	M-n200-k17	12	1, 154, 83, 9, 115, 200, 126, 46, 175, 47, 1
M	M-n200-k17	13	1, 103, 52, 10, 104, 162, 72, 136, 36, 137, 66, 67, 1
M	M-n200-k17	14	1, 5, 156, 140, 188, 26, 171, 40, 57, 187, 24, 68, 1
M	M-n200-k17	15	1, 112, 110, 178, 30, 25, 164, 135, 56, 166, 131, 55, 196, 150, 180, 1
M	M-n200-k17	16	1, 13, 185, 77, 197, 117, 78, 4, 159, 69, 81, 151, 122, 1
M	M-n200-k17	17	1, 147, 168, 133, 70, 102, 163, 32, 191, 128, 53, 107, 195, 8, 89, 1
P	P-n101-k4	1	1, 90, 7, 95, 96, 98, 93, 60, 97, 100, 94, 99, 38, 101, 43, 44, 15, 45, 39, 87, 18, 85, 62, 17, 92, 86, 6, 61, 1
P	P-n101-k4	2	1, 28, 29, 13, 77, 51, 2, 70, 71, 31, 21, 52, 10, 82, 34, 80, 4, 78, 69, 81, 25, 30, 79, 35, 36, 72, 67, 66, 1
P	P-n101-k4	3	1, 54, 27, 55, 5, 56, 26, 40, 68, 24, 57, 76, 23, 75, 73, 74, 22, 41, 59, 14, 88, 3, 58, 16, 42, 1
P	P-n101-k4	4	1, 53, 19, 84, 9, 83, 49, 8, 89, 32, 11, 33, 91, 64, 65, 12, 63, 20, 50, 37, 48, 47, 46, 1
P	P-n16-k8	1	1, 6, 8, 10, 1
P	P-n16-k8	2	1, 2, 11, 1
P	P-n16-k8	3	1, 16, 13, 1
P	P-n16-k8	4	1, 15, 4, 1
P	P-n16-k8	5	1, 5, 12, 1
P	P-n16-k8	6	1, 14, 9, 1
P	P-n16-k8	7	1, 7, 1
P	P-n16-k8	8	1, 3, 1
P	P-n20-k2	1	1, 11, 5, 12, 16, 13, 4, 19, 18, 9, 3, 1
P	P-n20-k2	2	1, 7, 20, 8, 6, 15, 17, 10, 14, 2, 1
P	P-n21-k2	1	1, 7, 21, 8, 6, 15, 18, 10, 14, 3, 2, 1

P	P-n21-k2	2	1, 17, 11, 5, 12, 16, 13, 4, 9, 19, 20, 1
P	P-n22-k2	1	1, 17, 2, 11, 5, 12, 16, 13, 4, 9, 19, 20, 1
P	P-n22-k2	2	1, 7, 21, 6, 22, 8, 3, 14, 10, 18, 15, 1
P	P-n40-k5	1	1, 28, 2, 23, 29, 32, 9, 27, 8, 24, 7, 1
P	P-n40-k5	2	1, 33, 3, 30, 21, 36, 37, 4, 25, 1
P	P-n40-k5	3	1, 12, 6, 39, 10, 17, 22, 35, 31, 11, 1
P	P-n40-k5	4	1, 19, 5, 18, 20, 14, 26, 15, 1
P	P-n40-k5	5	1, 13, 38, 16, 34, 40, 1
P	P-n45-k5	1	1, 33, 12, 17, 3, 30, 21, 36, 37, 4, 1
P	P-n45-k5	2	1, 6, 39, 10, 22, 35, 31, 11, 40, 34, 1
P	P-n45-k5	3	1, 28, 2, 23, 29, 32, 9, 27, 8, 24, 1
P	P-n45-k5	4	1, 7, 15, 19, 14, 26, 25, 44, 1
P	P-n45-k5	5	1, 13, 18, 38, 16, 45, 43, 20, 41, 42, 5, 1
P	P-n50-k10	1	1, 14, 20, 15, 12, 1
P	P-n50-k10	2	1, 17, 24, 44, 42, 43, 2, 23, 1
P	P-n50-k10	3	1, 34, 50, 25, 19, 33, 1
P	P-n50-k10	4	1, 6, 48, 22, 37, 38, 1
P	P-n50-k10	5	1, 27, 39, 11, 32, 1
P	P-n50-k10	6	1, 46, 28, 30, 49, 21, 1
P	P-n50-k10	7	1, 18, 13, 41, 45, 4, 1
P	P-n50-k10	8	1, 5, 40, 10, 26, 1
P	P-n50-k10	9	1, 35, 47, 8, 36, 9, 16, 1
P	P-n50-k10	10	1, 7, 3, 31, 29, 1
P	P-n50-k7	1	1, 18, 41, 13, 40, 10, 33, 1
P	P-n50-k7	2	1, 34, 2, 29, 23, 43, 42, 44, 24, 1
P	P-n50-k7	3	1, 27, 8, 36, 9, 47, 35, 28, 14, 20, 1
P	P-n50-k7	4	1, 49, 22, 48, 37, 38, 21, 16, 6, 1
P	P-n50-k7	5	1, 45, 4, 17, 50, 25, 19, 26, 1
P	P-n50-k7	6	1, 12, 15, 39, 11, 32, 1
P	P-n50-k7	7	1, 5, 46, 30, 31, 3, 7, 1
P	P-n51-k10	1	1, 7, 15, 26, 25, 1
P	P-n51-k10	2	1, 33, 12, 39, 17, 51, 22, 1
P	P-n51-k10	3	1, 49, 9, 27, 32, 29, 23, 1
P	P-n51-k10	4	1, 48, 5, 19, 1
P	P-n51-k10	5	1, 3, 24, 8, 44, 1
P	P-n51-k10	6	1, 2, 4, 37, 36, 21, 30, 1
P	P-n51-k10	7	1, 14, 42, 20, 43, 41, 1
P	P-n51-k10	8	1, 47, 13, 18, 38, 50, 28, 1
P	P-n51-k10	9	1, 10, 35, 31, 11, 40, 1
P	P-n51-k10	10	1, 6, 45, 16, 46, 34, 1
P	P-n55-k10	1	1, 3, 7, 52, 18, 41, 1
P	P-n55-k10	2	1, 8, 16, 6, 38, 21, 37, 1
P	P-n55-k10	3	1, 49, 48, 22, 29, 1
P	P-n55-k10	4	1, 27, 47, 35, 53, 9, 55, 1
P	P-n55-k10	5	1, 45, 33, 10, 26, 51, 1
P	P-n55-k10	6	1, 4, 17, 50, 25, 19, 1
P	P-n55-k10	7	1, 13, 40, 11, 39, 32, 1
P	P-n55-k10	8	1, 34, 2, 23, 43, 42, 44, 24, 1
P	P-n55-k10	9	1, 5, 46, 28, 14, 30, 31, 1
P	P-n55-k10	10	1, 36, 20, 15, 54, 12, 1
P	P-n55-k7	1	1, 35, 47, 9, 8, 36, 20, 55, 14, 28, 53, 1
P	P-n55-k7	2	1, 49, 22, 48, 37, 38, 21, 16, 6, 1

P	P-n55-k7	3	1, 54, 15, 12, 39, 11, 32, 1
P	P-n55-k7	4	1, 34, 2, 23, 43, 42, 44, 24, 17, 25, 1
P	P-n55-k7	5	1, 4, 45, 33, 10, 40, 26, 51, 19, 50, 1
P	P-n55-k7	6	1, 27, 13, 41, 18, 52, 7, 1
P	P-n55-k7	7	1, 5, 46, 30, 31, 3, 29, 1
P	P-n55-k8	1	1, 40, 11, 39, 32, 1
P	P-n55-k8	2	1, 46, 30, 49, 48, 37, 38, 21, 16, 6, 1
P	P-n55-k8	3	1, 41, 45, 33, 51, 19, 26, 10, 1
P	P-n55-k8	4	1, 2, 44, 43, 42, 24, 1
P	P-n55-k8	5	1, 31, 3, 22, 29, 23, 1
P	P-n55-k8	6	1, 18, 52, 7, 34, 17, 50, 25, 4, 13, 1
P	P-n55-k8	7	1, 8, 54, 36, 55, 20, 15, 12, 1
P	P-n55-k8	8	1, 27, 35, 47, 9, 53, 14, 28, 5, 1
P	P-n60-k10	1	1, 55, 14, 21, 38, 48, 37, 60, 1
P	P-n60-k10	2	1, 8, 59, 39, 11, 32, 1
P	P-n60-k10	3	1, 13, 40, 41, 4, 50, 25, 1
P	P-n60-k10	4	1, 46, 30, 58, 16, 6, 49, 31, 1
P	P-n60-k10	5	1, 45, 33, 10, 51, 26, 56, 1
P	P-n60-k10	6	1, 36, 20, 15, 54, 12, 1
P	P-n60-k10	7	1, 3, 23, 29, 22, 28, 1
P	P-n60-k10	8	1, 2, 44, 43, 42, 57, 24, 19, 1
P	P-n60-k10	9	1, 5, 35, 47, 53, 9, 1
P	P-n60-k10	10	1, 27, 18, 52, 7, 34, 17, 1
P	P-n60-k15	1	1, 46, 30, 6, 16, 28, 1
P	P-n60-k15	2	1, 35, 47, 53, 14, 1
P	P-n60-k15	3	1, 59, 11, 39, 1
P	P-n60-k15	4	1, 36, 54, 12, 1
P	P-n60-k15	5	1, 24, 57, 42, 43, 44, 1
P	P-n60-k15	6	1, 49, 22, 48, 37, 1
P	P-n60-k15	7	1, 5, 2, 50, 25, 1
P	P-n60-k15	8	1, 20, 55, 38, 21, 1
P	P-n60-k15	9	1, 18, 13, 41, 4, 1
P	P-n60-k15	10	1, 9, 15, 60, 1
P	P-n60-k15	11	1, 45, 51, 19, 26, 56, 1
P	P-n60-k15	12	1, 7, 52, 17, 34, 1
P	P-n60-k15	13	1, 3, 33, 32, 1
P	P-n60-k15	14	1, 31, 23, 29, 58, 1
P	P-n60-k15	15	1, 27, 8, 40, 10, 1
P	P-n65-k10	1	1, 49, 48, 22, 29, 62, 1
P	P-n65-k10	2	1, 35, 47, 53, 28, 14, 55, 9, 1
P	P-n65-k10	3	1, 2, 44, 23, 65, 43, 42, 57, 1
P	P-n65-k10	4	1, 8, 36, 39, 59, 11, 32, 1
P	P-n65-k10	5	1, 30, 6, 38, 37, 61, 21, 16, 58, 1
P	P-n65-k10	6	1, 46, 31, 3, 63, 7, 40, 1
P	P-n65-k10	7	1, 45, 33, 10, 26, 51, 19, 56, 1
P	P-n65-k10	8	1, 52, 17, 34, 64, 24, 50, 25, 4, 1
P	P-n65-k10	9	1, 54, 20, 15, 60, 12, 1
P	P-n65-k10	10	1, 18, 41, 13, 27, 5, 1
P	P-n70-k10	1	1, 5, 69, 31, 3, 29, 63, 1
P	P-n70-k10	2	1, 8, 36, 20, 60, 15, 54, 40, 1
P	P-n70-k10	3	1, 46, 49, 48, 22, 70, 37, 6, 1
P	P-n70-k10	4	1, 2, 44, 23, 65, 43, 42, 24, 57, 1

P	P-n70-k10	5	1, 18, 41, 13, 12, 32, 1
P	P-n70-k10	6	1, 68, 35, 28, 53, 47, 9, 1
P	P-n70-k10	7	1, 27, 59, 11, 39, 66, 67, 1
P	P-n70-k10	8	1, 30, 14, 55, 58, 16, 38, 21, 61, 62, 1
P	P-n70-k10	9	1, 7, 52, 17, 34, 64, 50, 25, 4, 1
P	P-n70-k10	10	1, 45, 33, 10, 26, 51, 19, 56, 1
P	P-n76-k4	1	1, 69, 34, 74, 63, 29, 75, 22, 37, 71, 61, 72, 70, 62, 23, 2, 44, 43, 65, 42, 57, 24, 64, 50, 1
P	P-n76-k4	2	1, 27, 13, 41, 18, 7, 52, 17, 4, 45, 33, 10, 40, 73, 32, 26, 51, 25, 19, 56, 1
P	P-n76-k4	3	1, 59, 11, 39, 66, 67, 12, 54, 15, 60, 20, 55, 9, 36, 8, 3, 1
P	P-n76-k4	4	1, 76, 5, 68, 35, 47, 53, 28, 46, 30, 14, 58, 16, 21, 38, 6, 48, 49, 31, 1
P	P-n76-k5	1	1, 18, 52, 4, 45, 33, 41, 13, 73, 40, 10, 26, 56, 51, 19, 25, 50, 1
P	P-n76-k5	2	1, 76, 69, 5, 68, 35, 47, 53, 46, 28, 14, 55, 9, 36, 8, 27, 1
P	P-n76-k5	3	1, 31, 75, 29, 62, 22, 48, 49, 30, 6, 38, 37, 70, 72, 61, 71, 21, 16, 1
P	P-n76-k5	4	1, 59, 11, 39, 54, 20, 15, 60, 12, 67, 66, 32, 1
P	P-n76-k5	5	1, 7, 17, 64, 24, 57, 2, 44, 42, 43, 65, 23, 63, 74, 34, 3, 58, 1
V	att-n48-k4	1	1, 34, 4, 23, 2, 10, 9, 17, 41, 1
V	att-n48-k4	2	1, 39, 32, 48, 6, 29, 3, 42, 25, 11, 45, 35, 26, 5, 1
V	att-n48-k4	3	1, 36, 30, 43, 18, 27, 20, 37, 7, 28, 8, 19, 44, 1
V	att-n48-k4	4	1, 15, 24, 12, 13, 16, 40, 38, 31, 46, 33, 21, 47, 22, 14, 1