

UNIVERSIDADE FEDERAL DO PARANÁ

LUCAS FERNANDES DE OLIVEIRA
MARCELA RIBEIRO DE OLIVEIRA

PROBLEMA DE ROTEAMENTO DE VEÍCULOS COM MÚLTIPLOS
DEPÓSITOS: UMA ABORDAGEM COEVOLUTIVA BASEADA EM ROTAS

CURITIBA PR
2017

LUCAS FERNANDES DE OLIVEIRA
MARCELA RIBEIRO DE OLIVEIRA

PROBLEMA DE ROTEAMENTO DE VEÍCULOS COM MÚLTIPLOS
DEPÓSITOS: UMA ABORDAGEM COEVOLUTIVA BASEADA EM ROTAS

Monografia de Trabalho de Graduação apresentada ao
Programa de Bacharelado em Ciência da Computação,
Departamento de Informática, Setor de Ciências Exatas,
Universidade Federal Do Paraná.

Área de concentração: *Ciência da Computação*.

Orientador: Prof. Dr. Eduardo Jaques Spinosa.

CURITIBA PR

2017

Agradecimentos

Diversas pessoas foram essenciais para chegarmos aonde estamos hoje, e a elas gostaríamos de agradecer.

Primeiramente a nossa amada família, por todos os momentos que nos apoiaram, faltam palavras para descrever o quão importantes vocês são para nós.

Aos nossos amigos, pela incrível insistência, cobrança e persistência para que concluíssemos esse trabalho. Em especial para aqueles que puderam comparecer a nossa apresentação, o seu apoio e presença foi fundamental.

Em especial gostaríamos de agradecer ao Giovane e ao Egon, pelas excelentes contribuições para o texto e ao Doutor Ricardo pela ajuda na elaboração da apresentação.

Também gostaríamos de agradecer ao nosso querido orientador pelo convite para desenvolvimento desse trabalho, pelas palavras motivadoras e por sempre ser prestativo e atencioso.

Por fim agradecemos ao nosso grande amigo Israel por todas as conversas e ideias compartilhadas.

Resumo

O Problema de Roteamento de Veículos (PRV) é uma generalização do conhecido Problema do Caixeiro Viajante (PCV) e uma representação para diversos problemas de logística. Nesse trabalho apresentamos um algoritmo alternativo para resolver uma das variações do PRV, o Problema de Roteamento de Veículos com Múltiplos Depósitos (PRVMD). Nesse algoritmo, propomos que a rota de cada veículo pode ser otimizada de forma independente e que a solução final é obtida combinando as soluções parciais independentes. Esse trabalho apresenta um conjunto de rotinas que compõe esse algoritmo, as interações entre elas e algumas sugestões de algoritmos para cada uma das rotinas.

Palavras-chave: algoritmos genéticos, roteamento de veículos, problemas de otimização.

Abstract

The Vehicle Routing Problem (VRP) is a generalization of the well-known Travelling Salesman Problem (TSP) and represents several logistics problems. We present an alternative algorithm to solve one of the variations of PRV, the Vehicle Routing Problem with Multiple Depots (VRPMD). In this algorithm, we propose that the route for each vehicle can be optimized independently and the final solution obtained compounding the partial independent solutions. We present a set of routines which compound this algorithm, the interactions among them and some suggestions of algorithms for each routine.

Keywords: genetic algorithms, vehicle routing, optimization problems.

Sumário

1	Introdução	12
2	Fundamentos	14
2.1	Problemas de otimização	14
2.2	Heurísticas	14
2.2.1	Meta-Heurísticas	15
2.3	Algoritmos Evolutivos	15
2.3.1	Algoritmos Genéticos	16
2.3.2	Algoritmos coevolutivos	18
3	Trabalhos Relacionados	19
3.1	Problema de roteamento de veículos com múltiplos depósitos	19
4	Proposta	21
4.1	Algoritmo	21
4.1.1	Preparação	21
4.1.2	Atribuição	22
4.1.3	Otimização	22
4.1.4	Consolidação	23
4.1.5	Fluxo	23
4.2	Implementação	24
4.2.1	Rotina de Otimização	24
4.2.2	Rotina de Atribuição	26
5	Análise Experimental	29
5.1	Experimentos	29
5.2	Resultados	30
5.2.1	Análise de Sensibilidade aos Parâmetros	30
5.2.2	Restrições de Custo e Capacidade	47
5.2.3	Custo Médio e Melhor Custo das Soluções	49
6	Conclusão	50
	Referências Bibliográficas	51
A	Instâncias	53
A.1	Instâncias propostas por Christofides e Eilon (1969):	53
A.1.1	Instância c01	53
A.1.2	Instância c02	54

A.1.3	Instância c03	54
A.1.4	Instância c04	55
A.1.5	Instância c05	55
A.1.6	Instância c06	56
A.1.7	Instância c07	56
A.2	Instâncias propostas por Gillett e Johnson (1976):	57
A.2.1	Instância g01	57
A.2.2	Instância g02	57
A.2.3	Instância g03	58
A.2.4	Instância g04	58
A.3	Instâncias propostas por Chao et al. (1993):	59
A.3.1	Instância ch01	59
A.3.2	Instância ch02	60
A.3.3	Instância ch03	60
A.3.4	Instância ch04	61
A.3.5	Instância ch05	61
A.3.6	Instância ch06	62
A.3.7	Instância ch07	62
A.3.8	Instância ch08	63
A.3.9	Instância ch09	63
A.3.10	Instância ch10	64
A.3.11	Instância ch11	64
A.3.12	Instância ch12	65
A.4	Instâncias propostas por Cordeau et al. (1997):	65
A.4.1	Instância co01	65
A.4.2	Instância co02	66
A.4.3	Instância co03	66
A.4.4	Instância co04	67
A.4.5	Instância co05	67
A.4.6	Instância co06	68
A.4.7	Instância co07	69
A.4.8	Instância co08	69
A.4.9	Instância co09	70
A.4.10	Instância co10	70

Lista de Figuras

2.1	Algoritmo Genético: Fluxograma Básico	16
4.1	Esquema do Algoritmo	21
4.2	Partição em Veículos	23
4.3	Operador de Mutação	24
4.4	Operador de Cruzamento	25
4.5	Operador de Cruzamento - Filho 1	25
4.6	Operador de Cruzamento - Filho 2	26
4.7	Multi-População: Topologia em Anel	26
4.8	Expansão e Compactação	28
5.1	Variação de parâmetros por custo médio - Instância c01	30
5.2	Variação de parâmetros por custo médio - Instância c02	31
5.3	Variação de parâmetros por custo médio - Instância c03	31
5.4	Variação de parâmetros por custo médio - Instância c04	32
5.5	Variação de parâmetros por custo médio - Instância c05	32
5.6	Variação de parâmetros por custo médio - Instância c06	33
5.7	Variação de parâmetros por custo médio - Instância c07	33
5.8	Variação de parâmetros por custo médio - Instância ch01	34
5.9	Variação de parâmetros por custo médio - Instância ch02	34
5.10	Variação de parâmetros por custo médio - Instância ch03	35
5.11	Variação de parâmetros por custo médio - Instância ch04	35
5.12	Variação de parâmetros por custo médio - Instância ch05	36
5.13	Variação de parâmetros por custo médio - Instância ch06	36
5.14	Variação de parâmetros por custo médio - Instância ch07	37
5.15	Variação de parâmetros por custo médio - Instância ch08	37
5.16	Variação de parâmetros por custo médio - Instância ch09	38
5.17	Variação de parâmetros por custo médio - Instância ch10	38
5.18	Variação de parâmetros por custo médio - Instância ch11	39
5.19	Variação de parâmetros por custo médio - Instância ch12	39
5.20	Variação de parâmetros por custo médio - Instância co01	40
5.21	Variação de parâmetros por custo médio - Instância co02	40
5.22	Variação de parâmetros por custo médio - Instância co03	41
5.23	Variação de parâmetros por custo médio - Instância co04	41
5.24	Variação de parâmetros por custo médio - Instância co05	42
5.25	Variação de parâmetros por custo médio - Instância co06	42
5.26	Variação de parâmetros por custo médio - Instância co07	43
5.27	Variação de parâmetros por custo médio - Instância co08	43
5.28	Variação de parâmetros por custo médio - Instância co09	44

5.29	Variação de parâmetros por custo médio - Instância co10	44
5.30	Variação de parâmetros por custo médio - Instância g01	45
5.31	Variação de parâmetros por custo médio - Instância g02	45
5.32	Variação de parâmetros por custo médio - Instância g03	46
5.33	Variação de parâmetros por custo médio - Instância g04	46
A.1	Mapa - Instância c01	53
A.2	Mapa - Instância c02	54
A.3	Mapa - Instância c03	54
A.4	Mapa - Instância c04	55
A.5	Mapa - Instância c05	55
A.6	Mapa - Instância c06	56
A.7	Mapa - Instância c07	56
A.8	Mapa - Instância g01	57
A.9	Mapa - Instância g02	58
A.10	Mapa - Instância g03	58
A.11	Mapa - Instância g04	59
A.12	Mapa - Instância ch01	59
A.13	Mapa - Instância ch02	60
A.14	Mapa - Instância ch03	60
A.15	Mapa - Instância ch04	61
A.16	Mapa - Instância ch05	61
A.17	Mapa - Instância ch06	62
A.18	Mapa - Instância ch07	62
A.19	Mapa - Instância ch08	63
A.20	Mapa - Instância ch09	63
A.21	Mapa - Instância ch10	64
A.22	Mapa - Instância ch11	64
A.23	Mapa - Instância ch12	65
A.24	Mapa - Instância co01	66
A.25	Mapa - Instância co02	66
A.26	Mapa - Instância co03	67
A.27	Mapa - Instância co04	67
A.28	Mapa - Instância co05	68
A.29	Mapa - Instância co06	68
A.30	Mapa - Instância co07	69
A.31	Mapa - Instância co08	69
A.32	Mapa - Instância co09	70
A.33	Mapa - Instância co10	70

Lista de Tabelas

5.1	Parâmetros dos Experimentos	29
5.2	Parâmetros dos Experimentos - Resultados finais	48
5.3	Média das Restrições Atendidas- Instâncias ch01-ch12	48
5.4	Média das Restrições Atendidas - Instâncias co01-co10	48
5.5	Média das Restrições Atendidas - Instâncias c01-c07	48
5.6	Média das Restrições Atendidas - Instâncias g01-g04	48
5.7	Custo por Instância- Instâncias ch01-ch12	49
5.8	Custo por Instância - Instâncias co01-co10	49
5.9	Custo por Instância - Instâncias c01-c07	49
5.10	Custo Médio por Instância - Instâncias g01-g04	49

Lista de Acrônimos

AE	Algoritmo Evolutivo
AG	Algoritmo Genético
AGMP	Algoritmo Genético Multi-Populacional
PCV	Problema do Caixeiro Viajante
PRV	Problema de Roteamento de Veículos
PRVMD	Problema de Roteamento de Veículos com Múltiplos Depósitos
BLI	Busca Local Iterativa

Capítulo 1

Introdução

Consideremos o seguinte problema. Uma pessoa é dona de uma empresa que realiza entregas utilizando caminhões. Esses caminhões estão distribuídos em diversos depósitos espalhados pela cidade. Todo dia esses caminhões devem realizar entregas. Para realizar uma entrega, o motorista do caminhão recebe uma rota, que define quais clientes ele deve atender com o seu caminhão e depois deve retornar ao mesmo depósito de onde partiu.

O dono dessa empresa tem um grande problema: definir qual é a rota que cada caminhão deve fazer. Seu objetivo é garantir que todos os clientes sejam atendidos e o custo de todas as rotas juntas seja o menor possível.

Entretanto há outros fatores a se considerar. O caminhão tem um tamanho, logo tem um número limitado de mercadorias que pode carregar. Os motoristas dos caminhões também têm um horário de expediente, logo a duração da sua rota não pode exceder o período da sua jornada de trabalho.

Essa é uma descrição informal do Problema de Roteamento de Veículos com Múltiplos Depósitos (PRVMD). A proposta deste trabalho é apresentar um algoritmo alternativo para resolver esse problema.

Esse é um problema que desafia os métodos de busca convencionais. O espaço de busca é muito grande para encontrar a resposta ótima em tempo hábil. Isso abre espaço para outras técnicas, como meta-heurísticas e soluções bioinspiradas.

Neste trabalho propomos um algoritmo alternativo para percorrer esse espaço de busca. Sua principal característica é dividir o problema até o nível de rotas e realizar a otimização de cada rota de forma independente.

Propomos a utilização do Algoritmo Genético (AG) para realizar a otimização das rotas. Entretanto, como o PRVMD é um problema com restrições, algumas alterações foram feitas no AG clássico para trabalhar com elas. Dentre estas alterações está uma estratégia que denominamos transferência de clientes, onde clientes de uma rota podem ser transferidos para outra baseado nas rotas em que estão seus vizinhos mais próximos.

Além do AG tradicional, uma abordagem multi-populacional também é proposta. E a partir disso diversas configurações de parâmetros foram testadas para analisar o impacto de cada um deles no desempenho do algoritmo e, conseqüentemente, no custo e qualidade da solução final.

Esse documento está organizado em 5 capítulos. No Capítulo 2, de fundamentos, apresentamos os conceitos fundamentais para entender o problema e sua complexidade, além dos fundamentos para a solução que propomos. Em trabalhos relacionados, no Capítulo 3, expomos o problema de forma mais precisa e apresentamos como outros pesquisadores tem abordado

o problema. Por fim, nos capítulos 4 e 5, apresentamos a proposta do nosso algoritmo e os experimentos realizados para validá-lo.

Capítulo 2

Fundamentos

2.1 Problemas de otimização

O problema de achar o valor máximo ou mínimo de uma função objetivo, possivelmente sujeito a um conjunto de restrições, é conhecido como problema de otimização. (Neto et al., 2009)

Segundo Engelbrecht (2007) um problema de otimização pode ser decomposto nas seguintes partes:

A **função objetivo** representa a quantidade a ser otimizada, isto é, a quantidade a ser minimizada ou maximizada. Seja f uma função objetivo. Então o máximo de f é o mínimo de $-f$. Alguns problemas, mais especificamente problemas de satisfação de restrição (CSP, *Constraint-satisfaction Problems*), não definem uma função de objetivo explícita. Em vez disso, o objetivo é encontrar a solução que satisfaça todo o conjunto de restrições.

Um **conjunto de incógnitas ou variáveis**, que afetam o valor da função objetivo. Se \mathbf{x} representa as incógnitas, também referidas como variáveis independentes, então $f(\mathbf{x})$ quantifica a qualidade da solução candidata, \mathbf{x} .

Um **conjunto de restrições**, que restringem os valores que podem ser atribuídos as incógnitas. A maioria dos problemas define pelo menos um conjunto de restrições de limite, que definem o domínio de valores para cada variável. As restrições podem, no entanto, ser mais complexas, excluindo certas soluções candidatas de serem consideradas soluções.

O objetivo do método de otimização é atribuir valores, dentro do espaço permitido, para as incógnitas de forma que a função objetivo seja otimizada e todas as restrições sejam satisfeitas. Para atingir esse objetivo, o algoritmo de otimização busca por uma solução no espaço de busca, S , de soluções candidatas. No caso de problemas com restrições, a solução é encontrada no espaço possível $F \subseteq S$. (Engelbrecht, 2007)

Algoritmos que resolvem problemas de otimização são denominados algoritmos de otimização. Tais algoritmos são métodos de busca, no qual o objetivo é encontrar uma solução para um problema de otimização, onde um certo valor é otimizado, possivelmente sujeito a um conjunto de restrições. (Engelbrecht, 2007)

2.2 Heurísticas

O nome heurística é derivado da palavra grega *heuriskein*, que significa descobrir. Neto et al. (2009) Heurísticas são métodos para encontrar soluções, tais métodos não podem ser provados matematicamente e por conseguinte não garantem uma solução ótima.

Dentre as grandes dificuldades dos problemas de otimização está o tamanho do espaço de busca e, dada essa condição, em muitos casos torna-se inviável percorrê-lo completamente em tempo hábil.

Em consequência da dificuldade acima apontada, em problemas dessa natureza, o uso de métodos exatos se torna bastante restrito. Por outro lado, na prática, em geral, é suficiente encontrar uma boa solução para o problema, ao invés do ótimo global, o qual somente pode ser encontrado após um considerável esforço computacional.(Souza, 2005)

Dessa forma então faz-se uso de heurísticas que procuram soluções próximas da ótima a um custo computacional aceitável.

2.2.1 Meta-Heurísticas

Uma meta-heurística é uma estratégia de busca, não específica para um determinado problema, que tenta explorar eficientemente o espaço das soluções viáveis desse problema. São algoritmos aproximados que incorporam mecanismos para evitar confinamento em mínimos ou máximos locais. Conhecimentos específicos do problema podem ser utilizados na forma de heurística para auxiliar no processo de busca.(Neto et al., 2009)

Existem diferentes métodos para se classificar meta-heurísticas, uma delas é considerar se a meta-heurística é inspirada ou não na natureza. Assim, dizemos que uma meta-heurística é bio-inspirada quando suas regras de busca tentam simular alguns aspectos do comportamento de seres vivos.(Neto et al., 2009)

Além disso dizemos que uma meta-heurística é baseada em população quando trabalha com um conjunto de candidatos a solução, que podem ser tratados, por exemplo, como indivíduos de uma população no caso dos algoritmos evolutivos.

2.3 Algoritmos Evolutivos

Computação evolutiva é uma sub área da inteligência artificial que utiliza modelos computacionais inspirados em processos evolutivos.

Algoritmos evolutivos são algoritmos de otimização meta-heurísticos populacionais que usam mecanismos inspirados na biologia como mutação, cruzamento, seleção natural, e sobrevivência do mais adaptado a fim de filtrar um conjunto de candidatos a solução iterativamente.(Weise, 2011)

De acordo com Yu e Gen (2010), os algoritmos evolutivos (AEs) têm três características principais:

- **Baseados em População:** Os AEs mantêm um grupo de soluções, chamado de população, para otimizar o problema de forma paralela. A população é um princípio básico do processo evolutivo.
- **Orientados à Aptidão:** Toda solução em uma população é chamada de indivíduo. Cada indivíduo tem um valor de aptidão que é calculado utilizando uma função. A aptidão mede a qualidade de uma solução, logo quanto maior a aptidão, melhor é a solução.
- **Orientados à Diversidade:** Os indivíduos serão submetidos a uma série de operações de variação para imitar mudanças genéticas, o que é fundamental para explorar o espaço de busca.

2.3.1 Algoritmos Genéticos

Algoritmos genéticos são provavelmente a classe de algoritmos evolutivos mais amplamente utilizada e aplicam-se também ao PRVMD.(Oliveira et al., 2016)

O Algoritmo Genético foi proposto inicialmente por John Holland em 1975.Holland (1975) Nesse algoritmo os indivíduos da população são representados através de uma cadeia genética. Cada iteração do algoritmo é chamada de uma geração e durante uma geração as soluções candidatas se reproduzem, utilizando operadores, gerando a população da próxima geração. Devido a forte relação do AG com conceitos biológicos a nomenclatura utilizada pelo mesmo é relacionada com termos encontrados na biologia.

Um AG típico utiliza três operadores: seleção, cruzamento e mutação, para direcionar a população, ao longo de uma série de etapas de tempo ou gerações, em direção à convergência no ótimo global.(Coley, 1999)

As características do indivíduo são representadas por um **cromossomo**, que também pode ser chamado de cadeia genética. Um cromossomo é representado por uma sequência, ou lista, de **genes**, isto é, **genoma**. Cada **gene**, que é considerado como a menor unidade de informação do algoritmo, faz referência a uma das variáveis do problema para as quais está se buscando o valor ideal. Um valor do domínio atribuído a um gene é chamado de **alelo**.

Um estágio importante no projeto de um algoritmo evolutivo é encontrar uma representação adequada de soluções candidatas (ou seja, cromossomos). A eficiência e complexidade do algoritmo de busca depende muito do esquema de representação.Engelbrecht (2007)

O AG é um algoritmo iterativo, isso significa que ele repete os mesmos passos até que uma condição de parada seja satisfeita. A figura 2.1 apresenta o fluxo básico de uma iteração do algoritmo, também chamada de geração. Normalmente a condição de parada é um número específico de gerações ou um número de gerações em que não há melhora nos indivíduos da população.

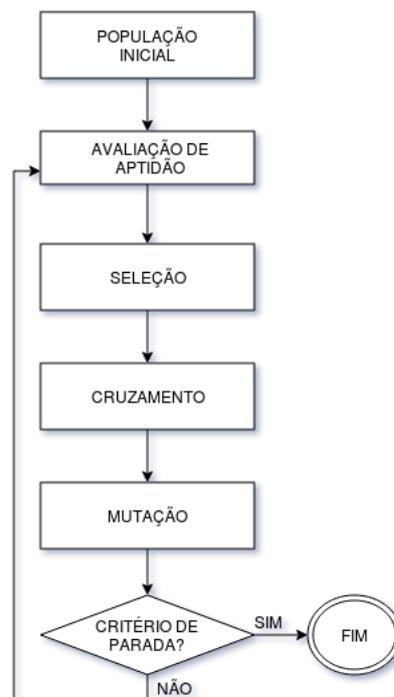


Figura 2.1: Algoritmo Genético: Fluxograma Básico

O AG funciona modificando a população de indivíduos a cada geração. A cada nova geração uma nova população é gerada utilizando a população da geração anterior. A única exceção neste processo é a população que compõe a primeira geração, chamada de população inicial.

2.3.1.1 Geração da população inicial

Como a primeira geração não possui a população da geração anterior, é necessário utilizar algum mecanismo para criá-la.

A forma como a população inicial é gerada afeta diretamente os resultados encontrados, comumente ela é gerada com indivíduos aleatórios. Este processo é feito atribuindo valores aleatórios presentes no domínio a cada um dos genes de cada cromossomo. Conforme aponta Engelbrecht (2007), o objetivo da seleção aleatória é garantir que a população inicial seja uma representação uniforme de todo o espaço de busca. Se regiões do espaço de busca não são cobertas pela população inicial, é provável que essas partes sejam negligenciadas pelo processo de busca.

Outro aspecto importante da população é a quantidade de indivíduos, denominada tamanho da população. Esse valor indica quantas explorações simultâneas estão ocorrendo no espaço de busca.

Encontrar o valor ideal para o tamanho da população é um desafio. Se a população contém poucos indivíduos, o algoritmo precisará de mais gerações para convergir, além de que pode vir a negligenciar partes do espaço de busca. Já em uma população com muitos indivíduos, o espaço de busca explorado em uma geração será maior, contudo, o custo para processar uma geração aumenta proporcionalmente ao tamanho da população.

2.3.1.2 Processando uma geração

O objetivo de uma geração é criar uma população com indivíduos mais aptos que os da geração anterior. Para isso, dois passos são executados.

Primeiramente os indivíduos da população anterior são avaliados. Após, os operadores do AG são aplicados.

Como descrito em 2.1, um dos ingredientes básicos de um problema de otimização é a função objetivo. No AG esta função é denominada função de aptidão. A avaliação de um indivíduo nada mais é do que aplicar a função de aptidão sobre o mesmo.

A função de aptidão deve ser precisa o suficiente a ponto de diferenciar indivíduos mais aptos / menos aptos. Caso contrário, uma boa solução pode ser desprezada no decorrer da execução do algoritmo.

Uma vez que os indivíduos da geração anterior foram avaliados, pode-se aplicar os operadores, conforme o fluxograma da figura 2.1.

O objetivo principal do operador de seleção é escolher as melhores soluções. Engelbrecht (2007) É ele que determinará em quais indivíduos os operadores genéticos, isto é, mutação e cruzamento, serão aplicados.

Existem vários operadores de seleção. No escopo deste trabalho abordaremos a seleção por roleta e a seleção por torneio.

A seleção por roleta é um tipo de seleção denominada seleção proporcional. Neste método de seleção a aptidão de cada indivíduo é normalizada, isto é, a aptidão de um indivíduo é a sua aptidão dividida pelo somatório da aptidão de todos os indivíduos da população.

A probabilidade de cada indivíduo ser selecionado é colocada em uma roleta. Dessa forma, cada indivíduo recebe uma fatia da roleta proporcional a sua aptidão normalizada.

Então roda-se a agulha da roleta, o indivíduo representado pela fatia que a agulha para é selecionado para cruzamento. Indivíduos com maior aptidão tem maiores chances de serem selecionados, já que ocupam fatias maiores da roleta.

Uma desvantagem desse operador é que se a roleta selecionar apenas os "melhores" indivíduos para serem aplicados os operadores genéticos, a diversidade da população da próxima geração diminui.

Na seleção por torneio uma quantidade n de indivíduos da população, sendo n o tamanho do torneio, é selecionada aleatoriamente para compor uma disputa. Nesta disputa então, é selecionado o melhor indivíduo, o de maior aptidão, para ser retornado pelo algoritmo. Para um cruzamento entre dois pais o torneio é executado duas vezes, uma para cada pai.

Dependendo do tamanho do torneio, o resultado da seleção pode ser bom ou ruim. Um torneio muito grande aumenta as chances dos melhores indivíduos serem selecionados para cruzamento, causando perda da diversidade na próxima geração. Já o torneio muito pequeno aumenta a probabilidade de indivíduos ruins serem selecionados.

O operador de cruzamento permite a criação de novos indivíduos. Ele é conhecido como operador genético principal e geralmente envolve dois indivíduos que reproduzem gerando um ou dois filhos que possuem características dos dois pais.

Esse operador combina o código genético dos indivíduos previamente selecionados para gerar filhos. Existem diversas formas de realizar essa combinação, cada uma caracteriza um operador diferente.

A mutação tem como objetivo introduzir diversidade na população. Fazendo isso, existe possibilidade de todo ponto no espaço de busca ser alcançado. Este operador é aplicado com uma certa probabilidade, geralmente pequena, a cada indivíduo resultante da operação de cruzamento.

Ao fim da fase de aplicação de operadores uma nova população terá sido gerada.

Uma vez que todos os operadores foram aplicados, uma nova população é criada, que substitui a anterior.

2.3.2 Algoritmos coevolutivos

Os algoritmos coevolutivos são uma classe de algoritmos evolutivos inspirados no processo de evolução simultânea envolvendo duas ou mais espécies. (Oliveira et al., 2016)

Esses algoritmos são divididos em duas categorias, dependendo da forma de interação das espécies, podendo ser cooperativos ou competitivos.

Na coevolução competitiva as espécies competem entre si, e dessa competição podem surgir indivíduos "melhores" em ambas as espécies. Engelbrecht (2007) Um exemplo é a simulação do comportamento "Predador-Presa", no cenário onde uma planta tenta sobreviver e insetos se alimentam dela, a planta pode evoluir a um estado onde possui uma superfície mais grossa. Para não morrerem, os insetos acabam evoluindo para obter mandíbulas mais fortes. (Engelbrecht, 2007)

Na coevolução cooperativa, a interação entre as espécies é benéfica ou na pior das hipóteses não causa nenhum prejuízo a nenhuma delas. Na natureza, essa forma de interação é denominada simbiose. Dessa forma, a força de uma espécie contribui para a sobrevivência da outra. (Engelbrecht, 2007)

No algoritmo coevolutivo cooperativo, cada espécie representa uma parte de um problema complexo que pode ser decomposto. Engelbrecht (2007) A aptidão de um indivíduo só pode ser obtida de sua interação com os outros indivíduos de sua população ou de outras populações. (Oliveira et al., 2016)

Capítulo 3

Trabalhos Relacionados

3.1 Problema de roteamento de veículos com múltiplos depósitos

O problema de roteamento de veículos com múltiplos depósitos, que é um variante do problema de roteamento de veículos clássico, pode ser definido da seguinte forma:

Dados:

- um conjunto de clientes, onde cada cliente possui uma demanda e um custo de serviço;
- um conjunto de depósitos;
- uma certa quantidade de veículos idêntica disponível por depósito;
- uma descrição de cada veículo, que consiste em uma capacidade de peso máxima que cada um pode transportar e em um custo máximo de serviços;
- e o custo de se deslocar entre todos os clientes e depósitos;

encontrar o conjunto de rotas de custo mínimo tal que:

- cada cliente deve ser atendido por exatamente um veículo;
- a demanda em uma rota não pode ser superior à capacidade do veículo que a percorre;
- a custo da rota não deve ser superior ao custo máximo do veículo que a percorre;
- e cada veículo deve começar e terminar seu percurso no mesmo depósito.

Uma rota é o percurso de um veículo, que é definida por uma lista ordenada, onde o primeiro elemento é o depósito de origem, os seguintes são os clientes que o veículo irá servir o último elemento é o depósito de origem novamente.

A demanda de uma rota é calculada somando a demanda de todos os clientes que fazem parte dela. Já o custo de uma rota é calculado somando o custo de serviço de cada cliente, mais o custo de se deslocar do primeiro para o segundo elemento, do segundo para o terceiro e assim sucessivamente até o fim da rota.

Esse problema pode ser representado por um grafo ponderado completo, onde os vértices são os clientes e os depósitos. O peso das arestas representa o custo de se deslocar de um vértice

a outro. As rotas representam circuitos no grafo, que são iniciados e concluídos no vértice que representa o depósito de origem.

Diferentes abordagens para resolver o PRVMD são encontradas na literatura. Dessa maneira, para melhor compreensão da abordagem proposta, a seguir apresentaremos aquelas nas quais a técnica utilizada é similar a deste trabalho.

Oliveira et al. (2016) propõem um algoritmo coevolutivo cooperativo para minimizar o custo das rotas do PRVMD. Na forma de decomposição do problema utilizada pelos autores, cada subproblema torna-se um PRV e dessa forma evolui independentemente no seu espaço de domínio. E, conforme apontam os autores, esta abordagem é fortemente adequada para ambientes paralelos.

Nessa abordagem os clientes são distribuídos entre os depósitos com base em sua distância dos depósitos e de sua distância do seu vizinho mais próximo. Uma vez que essa distribuição está completa, cada depósito torna-se uma população e, conseqüentemente um PRV independente de todos os outros depósitos. Os conjuntos de rotas formados por clientes atribuídos a cada depósito são soluções parciais para o problema, portanto representam um indivíduo.

Oliveira et al. (2016) também propõem uma estratégia de evolução paralela com um genótipo de tamanho variável associado a operadores de busca local.

Vidal et al. (2014) trabalham com uma abordagem diferente para atribuição inicial. Em seu trabalho, os autores propõem, utilizando programação dinâmica, encontrar o que os mesmos definem como rotação ideal. Isto é, gerenciar otimamente as opções de veículo, depósito e primeiro cliente visitado em uma rota.

Os autores também propõem um algoritmo de divisão que segmenta um genoma, escolhe veículo e depósitos e ainda rotaciona uma rota.

Como prova de conceito, essas metodologias são integradas e testadas dentro de duas meta-heurísticas: Uma Busca Local Iterativa (BLI) simples, com múltiplos pontos de partida e uma mais elaborada, chamada Busca Genética Híbrida com Controle de Diversidade Avançado Vidal et al. (2014).

Capítulo 4

Proposta

Algoritmos evolutivos foram utilizados na resolução do PRV e suas variações, alguns exemplos são encontrados em Oliveira et al. (2016), Jozefowicz et al. (2009) e em alguns casos, para o problema do caixeiro-viajante Ulder et al. (1991). Como foi apresentado por outros trabalhos sobre o problema Oliveira et al. (2016), esse problema pode ser dividido em subproblemas, o que reduz o espaço de busca, mas restringe as soluções encontradas. Contudo, mesmo realizando essa divisão, bons resultados ainda podem ser encontrados.

A proposta deste trabalho é resolver o PRVMD subdividindo o problema em partes menores que colaboram entre si para chegar a solução final. Essa abordagem é inspirada nos conceitos de algoritmos evolutivos e no processo de coevolução cooperativa.¹

4.1 Algoritmo

Para resolver o PRVMD propomos uma arquitetura para o algoritmo que possui duas rotinas, uma de atribuição e uma de otimização. Embora essas duas sejam as rotinas principais, existem duas rotinas secundárias, uma de preparação e outra de consolidação, responsáveis por ler e formatar a entrada, organizar os dados para a execução da atribuição e otimização e, montar a solução final do problema. A Figura 4.1 apresenta um esquema da arquitetura do algoritmo.

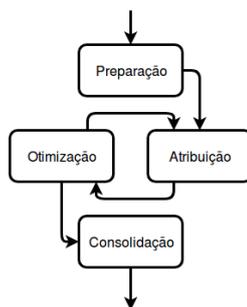


Figura 4.1: Esquema do Algoritmo

4.1.1 Preparação

A preparação é a rotina na qual o algoritmo é aprontado para executar. O objetivo dessa rotina é ler a entrada, verificar se a mesma está correta, e representá-la internamente de forma que as outras rotinas entendam.

¹A implementação dessa proposta está disponível em: <https://github.com/mro15/mdvrp-coevolutionary-approach>

Para representar o problema, as seguintes estruturas foram utilizadas:

- Um grafo completo, que tem como vértices clientes e depósitos, no qual as arestas contém o custo de se deslocar de um vértice para outro. Por questões de eficiência, o custo das arestas é representado por uma matriz $N \times N$ onde N é o número de vértices.
- Uma lista de veículos, onde cada veículo contém as informações sobre as restrições do problema (duração máxima da rota, capacidade máxima do veículo), a quantidade de depósitos existentes e a qual depósito esse veículo pertence. Essa lista tem o tamanho máximo de veículos possível: a quantidade de depósitos multiplicada pelo máximo por depósito.

4.1.2 Atribuição

A rotina de atribuição busca definir quais clientes os veículos devem atender. Devido às restrições do próprio problema, um cliente não deve ser atendido por mais de um veículo. Essa rotina garante que essa restrição será satisfeita, atribuindo um cliente a apenas um veículo.

Além de atribuir, essa rotina se encarrega de transferir clientes entre veículos, ou seja, remove um cliente de um veículo e o atribui a outro. Essa operação garante que o cliente continua sendo atendido por apenas um veículo.

A atribuição é feita alterando as estruturas de dados da seguinte forma:

- No grafo, cada vértice tem um atributo que identifica a qual veículo ele pertence. Vértices que representam depósitos não são atribuídos a nenhum veículo.
- Cada veículo mantém um conjunto de vértices de clientes que deve atender.

4.1.3 Otimização

A rotina de otimização busca encontrar a melhor permutação de clientes para um veículo, em outras palavras, a ordem que o veículo deve atender os clientes que a ele foram atribuídos de forma a minimizar o custo total da rota.

Durante a otimização, os veículos se mantêm de forma isolada. Isto é, as rotas de cada veículo são construídas de forma independente e, além disso, a construção de uma rota não interfere nos resultados alcançados pela rota de outro veículo.

Segundo a definição do problema, a única interferência entre veículos é a restrição que indica que um cliente não pode ser atendido por mais de um veículo. Essa restrição já foi garantida pela rotina de atribuição, e como nessa rotina não existe alteração no conjunto de clientes a serem atendidos, as rotas podem ser otimizadas independentemente.

Observe que, em relação ao problema original (PRVMD), a otimização de cada rota tem um espaço de busca muito mais restrito, por duas razões. Cada veículo possui um conjunto restrito (subconjunto) de clientes, e apenas um depósito para o qual o veículo foi atribuído.

Como o veículo deve sair e voltar ao depósito, percorrendo os clientes em uma determinada ordem sem repetir clientes, o problema se transforma em encontrar um circuito mínimo em um grafo completo, já que o grafo induzido por vértices em um grafo completo, também é completo. Esse problema é exatamente o mesmo do conhecido Problema do Caixeiro Viajante.

Portanto, a rotina de otimização, na realidade está buscando a solução do problema do caixeiro viajante, onde as cidades são os clientes atribuídos ao veículo e o depósito a qual ele pertence.

Ao final da rotina de otimização, as estruturas de dados recebem as seguintes modificações:

- Cada veículo mantém a melhor solução encontrada até agora;
- Cada veículo informa se as restrições de capacidade e de custo foram satisfeitas.

4.1.4 Consolidação

A rotina de consolidação verifica o resultado da rotina de otimização e forma a solução completa do problema. Juntando a melhor solução encontrada por cada veículo e calculando o custo de cada rota, a solução do problema está completa. Entretanto, além disso, essa rotina também verifica e contabiliza quais restrições foram satisfeitas em cada rota. A Figura 4.2 apresenta uma visão mais detalhada da arquitetura do algoritmo.

Depois que todas as junções e verificações são feitas o algoritmo termina e devolve o conjunto de rotas que serão percorridas pela frota de veículos.

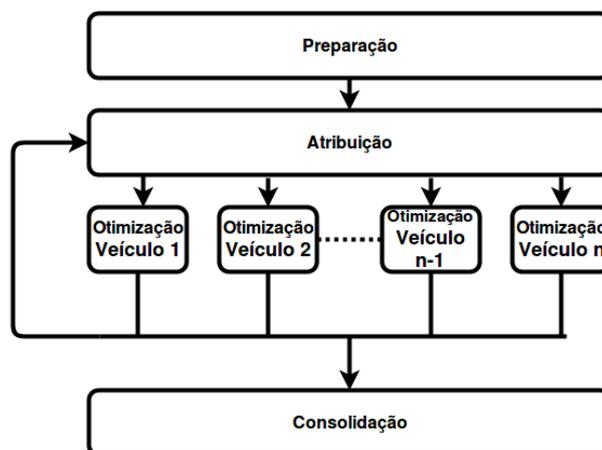


Figura 4.2: Partição em Veículos

4.1.5 Fluxo

As rotinas de preparação e de consolidação são executadas apenas uma vez, no começo e no fim do algoritmo respectivamente. As rotinas de atribuição e de otimização se alternam na execução, a primeira buscando encontrar a melhor divisão de clientes enquanto a outra busca encontrar a melhor ordenação de clientes. Assumimos que essas fases devem se alternar entre si para melhorar a solução encontrada.

Nota-se que se a rotina de atribuição retornasse a divisão ótima, ela teria que ser executada apenas uma vez, entretanto, não conhecemos um algoritmo que realiza essa divisão de forma perfeita, e consideramos que se a rotina de atribuição e otimização executassem de forma alternada seria mais produtivo, uma vez que a rotina de atribuição realiza a interação entre rotas, que contribui para cooperação na construção da solução.

De certa forma, PRVMD foi transformado em dois problemas de otimização. O primeiro é resolvido pela rotina de atribuição, que busca encontrar a melhor atribuição de clientes a rotas. A rotina de otimização resolve o PCV, que é usado para verificar quão boa aquela atribuição de clientes é.

4.2 Implementação

É possível implementar tanto a rotina de otimização quanto a rotina de atribuição de diversas formas. Busca tabu, otimização por colônia de formigas e algoritmos genéticos são exemplos de métodos que foram utilizados para resolver o PRV por Cordeau et al. (1997), Dorigo e Gambardella (1997) e Ulder et al. (1991) respectivamente.

Para a rotina de atribuição, também é possível imaginar que há diversas maneiras diferentes de dividir os clientes, e que o método utilizado tem impacto direto na solução final.

Partimos do princípio que os métodos de otimização e de atribuição devem trabalhar em conjunto, assim escolhemos como alternativa de otimização o AG e um método que utiliza uma abordagem coevolutiva cooperativa para realizar a atribuição.

4.2.1 Rotina de Otimização

O AG é um algoritmo com nomenclatura e conceitos próprios, para utilizá-lo como método de otimização temos que remodelar o problema, ou seja, traduzir os termos e conceitos do problemas para a nomenclatura do AG.

Um veículo se transforma em uma espécie. Cada espécie tem um genoma que define quais os possíveis indivíduos que podem pertencer àquela espécie, esse genoma são os clientes e depósitos. Uma rota corresponde a um indivíduo do AG, uma vez que representa uma possível solução para o problema.

Então, agora temos um conjunto de espécies que colaboram entre si com o objetivo de obter o melhor resultado. Durante a rotina de otimização não há interação entre as espécies. Essa interação é feita na rotina de atribuição e na de consolidação.

4.2.1.1 Operadores

Utilizamos duas alternativas para o operador de seleção, roleta e torneio. Como em experimentos iniciais o operador torneio se mostrou muito mais efetivo que o operador roleta, nos experimentos apresentados, apenas o operador torneio é utilizado na sequencia deste trabalho.

Para o operador de mutação utilizamos o seguinte método (Figura 4.3):

- Escolhem-se aleatoriamente duas posições da cadeia genética;
- Os genes das posições escolhidas são trocados.

Esse operador garante que não existem genes repetidos no mesmo indivíduo, o que do contrário violaria a restrição que obriga cada cliente ser visitado exatamente uma vez.

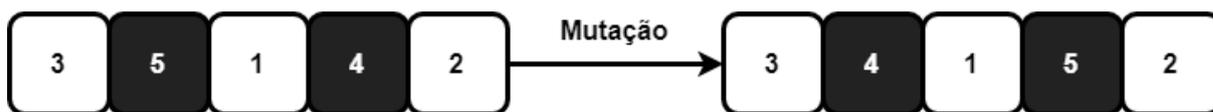


Figura 4.3: Operador de Mutaçao

Para o operador de cruzamento, um operador similar ao corte de um ponto é utilizado. Dado um par de indivíduos pais selecionados para realizar o cruzamento, o primeiro indivíduo denominaremos pai 1 e o segundo indivíduo denominaremos pai 2. A operação de cruzamento resultará em dois novos indivíduos, o primeiro denominaremos filho 1 e o segundo filho 2. Assim, o processo realizado é o seguinte:

- Sorteia-se uma posição aleatória na cadeia genética;
- O filho 1, manterá a mesma ordenação da cadeia genética do pai 1 antes dessa posição sorteada. E, a partir dessa posição os genes remanescentes do pai 1 são concatenados ao código genético do filho 1 na ordem em que aparecem no pai 2;
- O filho 2, manterá a mesma ordenação da cadeia genética do pai 2 antes dessa posição sorteada. E, a partir dessa posição os genes remanescentes do pai 2 são concatenados ao código genético do filho 2 na ordem em que aparecem no pai 1.

Este processo está detalhado nas figuras 4.4, 4.5 e 4.6. Como os genes restantes são apenas remanejados, não existe a possibilidade de um cliente aparecer mais de uma vez na cadeia, nem desaparecer, logo a restrição de visitar cada cliente apenas uma vez ainda é mantida.

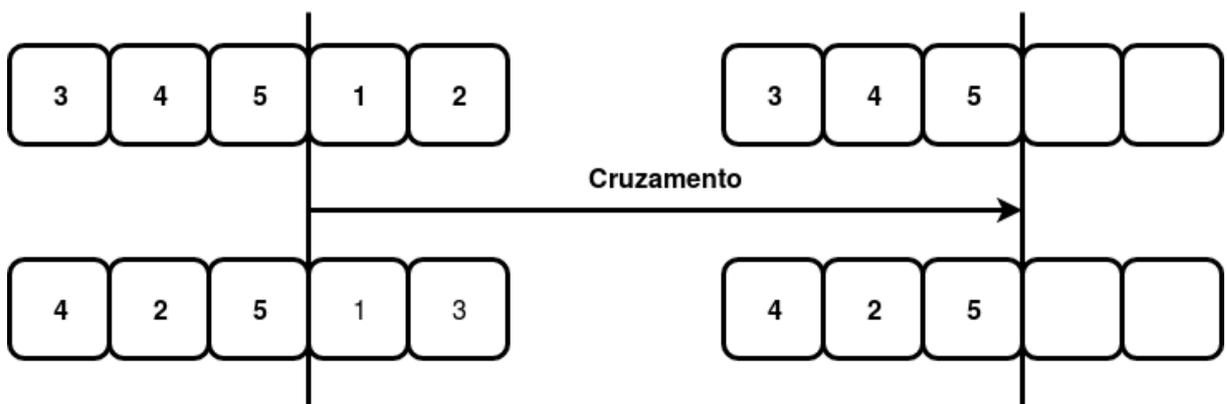


Figura 4.4: Operador de Cruzamento

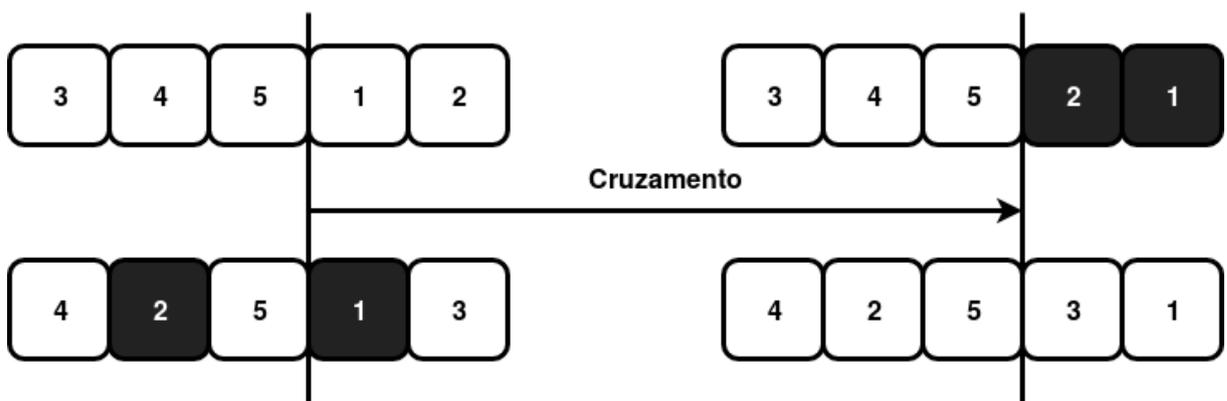


Figura 4.5: Operador de Cruzamento - Filho 1

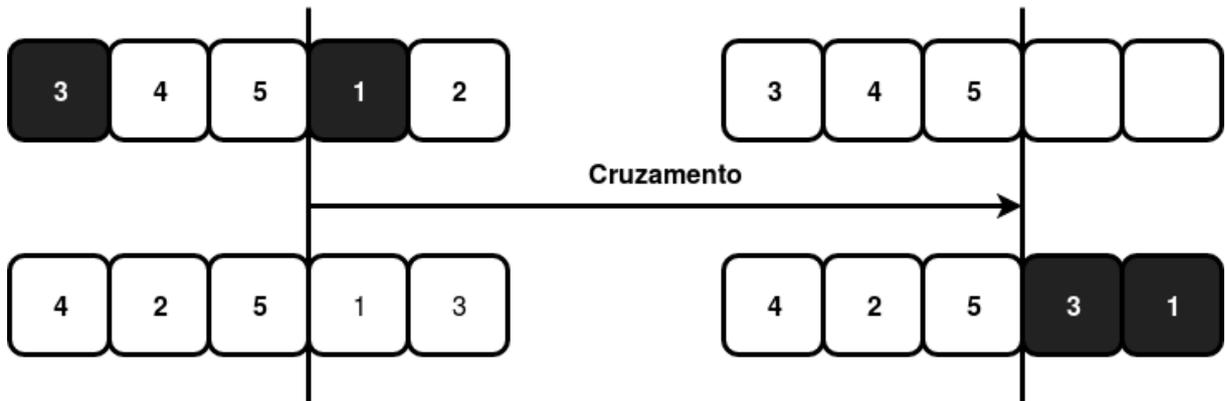


Figura 4.6: Operador de Cruzamento - Filho 2

4.2.1.2 Algoritmo Genético Multi-Populacional

Além da versão mais simples do AG, apresentada na seção anterior, uma versão multi-populacional também foi utilizada. Nessa versão, uma espécie possui diversas populações, ao invés de apenas uma.

Nos experimentos consideramos a variação granularidade grossa. Essa variação simula o isolamento geográfico. Cruzamentos ocorrem apenas entre indivíduo da mesma população e eventualmente ocorre a migração de indivíduos.

A cada certo número de gerações ocorre uma migração entre as populações. Indivíduos de uma população são movidos para outra população, possivelmente trazendo trechos da cadeia genética que não existem na população atual, aumentando variabilidade genética.

Na abordagem proposta nesse trabalho, exatamente um indivíduo de cada população migra. Utilizou-se uma topologia de anel entre as populações, e a migração ocorre sempre no mesmo sentido, como descrito pela Figura ???. Em outras palavras, cada população é 'adjacente' a outras duas, uma sempre recebe um indivíduo migrante da atual, e a outra sempre envia um migrante para a atual.

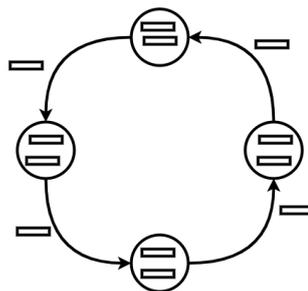


Figura 4.7: Multi-População: Topologia em Anel

4.2.2 Rotina de Atribuição

4.2.2.1 Atribuição Inicial

A atribuição inicial define o método pelo qual os clientes são inicialmente distribuídos entre os veículos, usado no momento que o algoritmo passa da rotina de preparação para a rotina de atribuição. Durante experimentos iniciais várias técnicas foram utilizadas, porém apenas uma será descrita aqui por ter obtido resultados melhores.

A técnica utilizada para definir a atribuição inicial primeiramente atribui os clientes ao depósito mais próximo.

Então utilizamos o seguinte algoritmo para definir qual será o cliente inicial de cada rota:

1. O primeiro cliente da rota 1 é o vértice mais distante do depósito.
2. Para definir o cliente inicial da rota N aplicam-se os seguintes passos:
 - (a) Todos os clientes possuem uma lista dos outros clientes ordenadas por distância. Inicia-se uma variável i com valor 1.
 - (b) Procura-se na lista de todos clientes iniciais já atribuídos o cliente não atribuído que aparece em todas as listas entre a primeira e a i -ésima posição.
 - (c) Se não existe um cliente não atribuído que não aparece em todas as sub-listas, incrementa-se o valor de i em 1 e repete o passo anterior.
3. Esse processo se repete até que todas as rotas tenham um cliente inicial.

Uma vez que cada rota tem um cliente inicial, cada rota, de forma alternada, tem atribuídos para si o cliente não atribuído a nenhuma outra rota mais próximo do cliente inicial pertencente ao depósito dos veículos em questão.

4.2.2.2 Transferência de clientes

Em métodos coevolutivos, a interação entre as espécies normalmente ocorre através de uma influência sobre a função de aptidão Engelbrecht (2007). Contudo, como observamos, a otimização de cada veículo não gera impacto sobre outro veículo, logo não faria sentido que a função de aptidão de uma espécie fosse afetada por outras espécies.

Entretanto, como modo de interação, temos a transferência de clientes entre veículos. Sob o ponto de vista evolutivo, temos uma alteração do conjunto de genes de uma espécie, portanto, é equivalente a se dizer que duas novas espécies são criadas ao se transferir um cliente de um veículo para outro.

Observe que ao alterar o conjunto de genes, todos os indivíduos passam a ser inválidos, ou seja, não representam mais a espécie, pois os genes que os formam são diferentes. Logo toda a população de indivíduos é afetada.

Ao invés de descartar todos os indivíduos, consideramos que a população existente pode ser corrigida, se adaptando ao novo código genético.

A espécie que perdeu um gene, deve remover de todos os indivíduos o gene que perdeu, e os genes adjacentes a esse se conectam. Essa operação nomeamos de compactação do indivíduo.

Já a espécie que ganhou um gene deve acrescentar a todos os indivíduos da população este novo gene. O mesmo é colocado em uma posição aleatória da cadeia genética. Essa operação recebeu o nome de expansão do indivíduo.

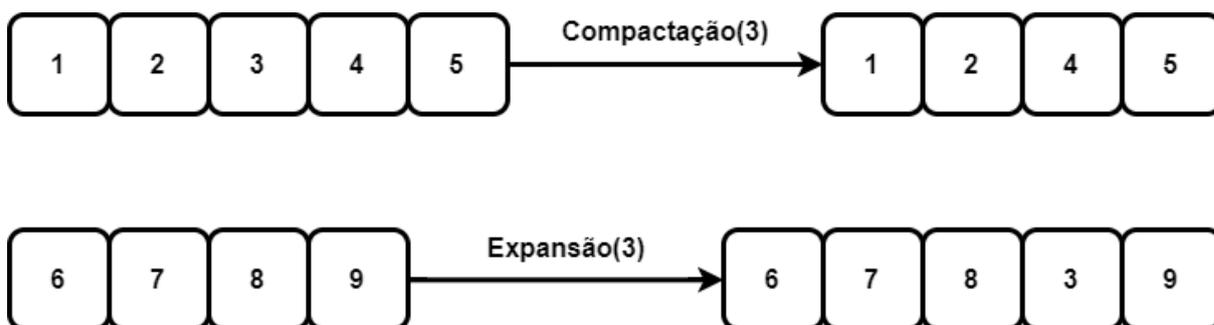


Figura 4.8: Expansão e Compactação

Além das configurações da rotina de otimização, a rotina de atribuição também deve ser configurada. Primeiramente, deve-se decidir como será feita a alternância entre rotina de otimização e a de atribuição. Definiu-se que depois de um número pré-determinado de gerações, que vêm a ser um parâmetro, assim como o período de migração do AGMP, será realizada a transferência de clientes, seguida da compactação e expansão dos indivíduos das populações envolvidas.

Para selecionar quais clientes serão transferidos e para onde, primeiramente identificam-se os clientes que se encontram em uma fronteira de rota. Estes são clientes que têm vizinhos próximos em outras rotas.

Com base na quantidade de vizinhos de rotas diferentes que estão próximos e nas restrições de capacidade do veículo atual e do veículo alvo da transferência, um valor denominado importância é calculado. Este valor simboliza o quanto seria benéfico ou prejudicial migrar este cliente da fronteira.

Este trabalho propõe dois métodos de escolha baseados no valor da importância. O primeiro seleciona apenas a migração de maior importância, realizando uma única transferência. O segundo método também escolhe a migração de maior importância, mas continua realizando as próximas transferências desde que tanto a população fonte quanto a população alvo do cliente ainda não tenham participado de nenhuma transferência nessa execução da rotina. Isso faz com que mais veículos sejam afetados por transferências, sendo que a população ou transferiu exatamente um cliente ou recebeu exatamente um cliente ou não participou de transferências.

Essa limitação de apenas uma transferência por veículo permite medir o impacto dessa transferência em relação ao estado anterior. Em outras palavras, é possível comparar as espécies anteriores com as novas e reverter o processo. Reverter a transferência poderia ser vantajoso, por exemplo, em casos onde as melhores soluções de ambas as espécies é pior que a das espécies ancestrais.

A definição de proximidade também é um parâmetro do método proposto. Ao invés de se considerar um valor numérico e absoluto de distância para todos os clientes, foi utilizada a quantidade de clientes mais próximos. Cada cliente seleciona K outros clientes e, se algum deles for de uma rota diferente da rota do cliente em questão, considera-se que este cliente está na fronteira de rotas.

A cada vez que essa rotina é executada, o número K é acrescido em 1, até um número máximo, que é um parâmetro do algoritmo.

Capítulo 5

Análise Experimental

5.1 Experimentos

Para validar o algoritmo proposto realizamos experimentos em um conjunto de 33 instâncias. As instâncias c01-c07 foram propostas por Christofides e Eilon (1969), as instâncias g01-g04 por Gillett e Johnson (1976), as instâncias ch01-ch12 por Chao et al. (1993) e por fim as instâncias co01-co10 foram propostas por Cordeau et al. (1997). Uma breve descrição de cada instância bem como o seu mapa com clientes e depósitos pode ser encontrada no apêndice A.¹

A tabela 5.1 descreve todos os parâmetros do algoritmo e apresenta os valores que foram utilizados para realizar os experimentos.

Parâmetros	Valores	Descrição
QtdIter	10; 100; 1000	Quantidade de iterações do método (gerações)
QtdInd	10; 50; 100	Quantidade de indivíduos por população
QtdPop	1; 2; 4	Quantidade de populações
PerMig	5; 10; 50	Período de migração (a cada quantas gerações)
TaxMut	0.0625; 0.125	Taxa de mutação
PerTrans	10; 20; 100	Período de transferência (a cada quantas gerações)
MinFront	3	Valor mínimo para definição da fronteira
MaxFront	7	Valor máximo para definição da fronteira

Tabela 5.1: Parâmetros dos Experimentos

O parâmetro QtdInd indica a quantidade de indivíduos por população, dessa forma se o parâmetro QtdPop for 2, e QtdInd for 100, o método utiliza um total de 200 indivíduos.

Os valores dos parâmetros MinFront e MaxFront foram definidos em experimentos preliminares. Para esses dois parâmetros, apenas um valor será considerado para reduzir a quantidade de combinações analisada neste capítulo.

O objetivo destes experimentos é demonstrar que essa abordagem é aplicável e encontrar quais são os melhores valores de cada parâmetro para utilizá-la. Para avaliar a aplicabilidade da abordagem são utilizadas duas métricas: o custo da solução encontrada e a quantidade de restrições atendidas.

¹A descrição completa de cada instância pode ser acessada em: <https://github.com/fboliveira/MDVRP-Instances>.

5.2 Resultados

Como pode-se observar na tabela 5.1, os experimentos foram realizados com uma variação de diversos parâmetros. Nas seções a seguir apresentamos alguns dos resultados mais relevantes encontrados.

Primeiramente são apresentados um conjunto de gráficos. Esse gráficos foram utilizados para definir qual seria a combinação ideal de parâmetros para as instâncias.

5.2.1 Análise de Sensibilidade aos Parâmetros

A seguir são apresentados gráficos que exibem o impacto da variação dos parâmetros sobre os resultados obtidos pelo algoritmo. Para cada instância é apresentado um gráfico comparativo de custo médio. O objetivo desses gráficos é avaliar o impacto de cada parâmetro individualmente.

São 5 parâmetros que são avaliados: tamanho da população, taxa de mutação, quantidade de gerações, período de migração e período de transferência. Nesses gráficos, os resultados das execuções são agrupados pelo valor do parâmetro em análise.

Exemplificando, os gráficos do parâmetro taxa de mutação com o valor 0.0625 possui a média do custo encontrado por **todos** os casos de teste que utilizaram o valor 0.0625 para o parâmetro de taxa de mutação, enquanto o valor 0.125 possui a média do custo encontrado por **todos** os casos de teste que utilizaram taxa de mutação com o valor 0.125. Ou seja, todos os outros parâmetros variam livremente, e apenas o impacto do valor da taxa de mutação é avaliado.

Essa mesma lógica foi utilizada para todos os parâmetros, dessa forma o impacto de cada parâmetro é avaliado, independente do valor dos outros parâmetros.

O impacto do valor de um parâmetro é medido através de duas métricas exibidas nos gráficos, a média dos custos encontrados em cada experimento e o desvio padrão.

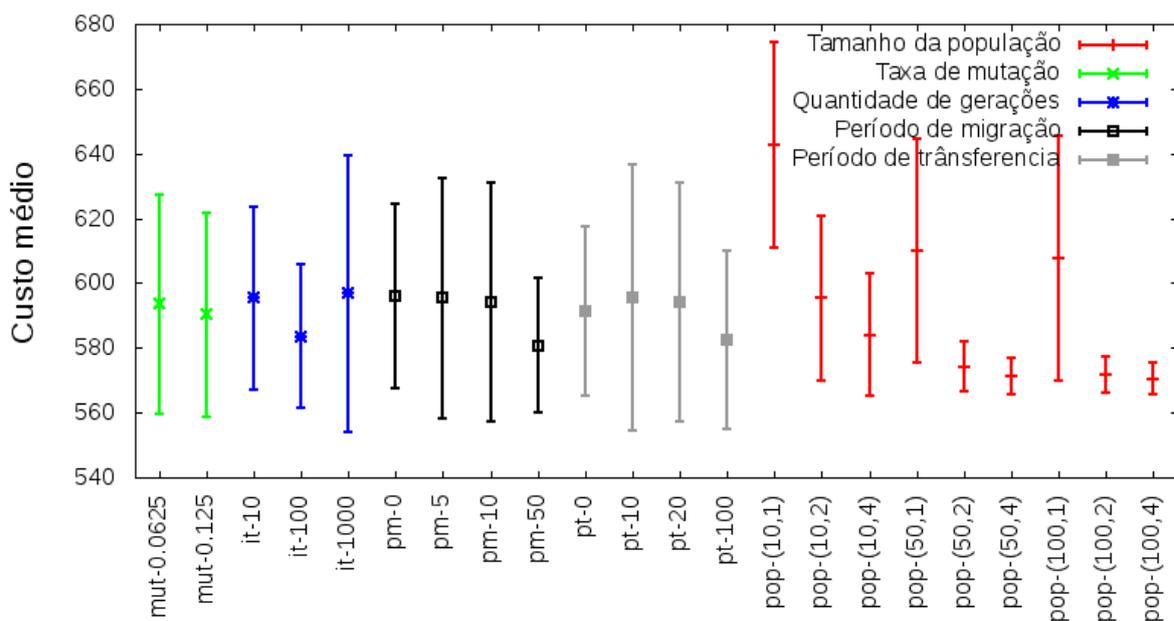


Figura 5.1: Variação de parâmetros por custo médio - Instância c01

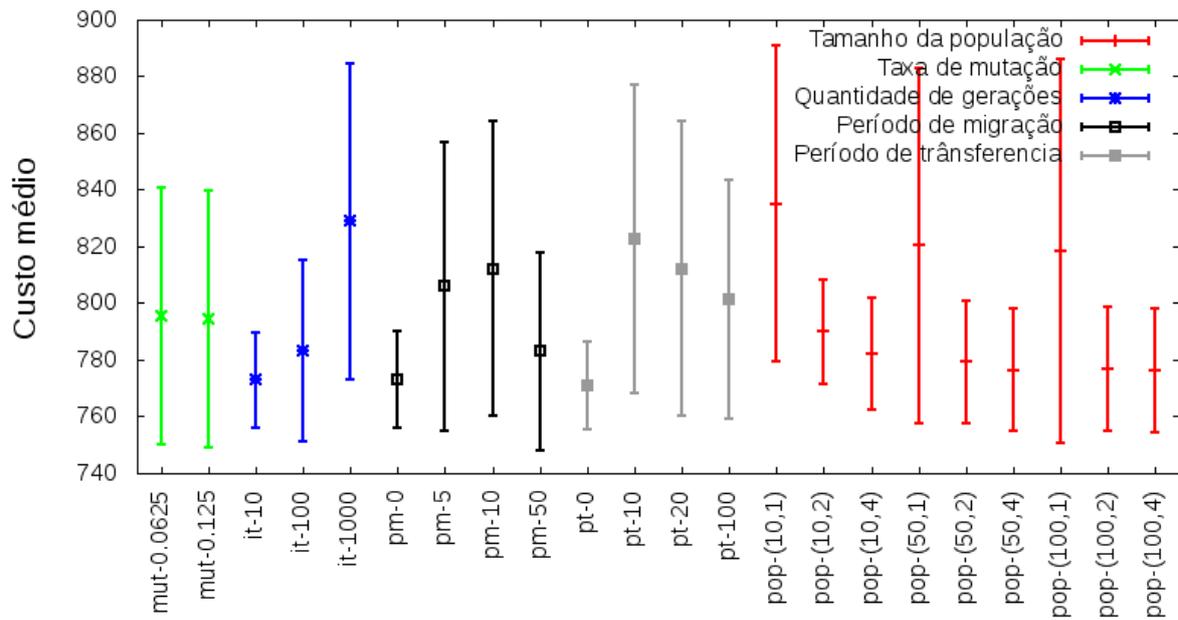


Figura 5.2: Variação de parâmetros por custo médio - Instância c02

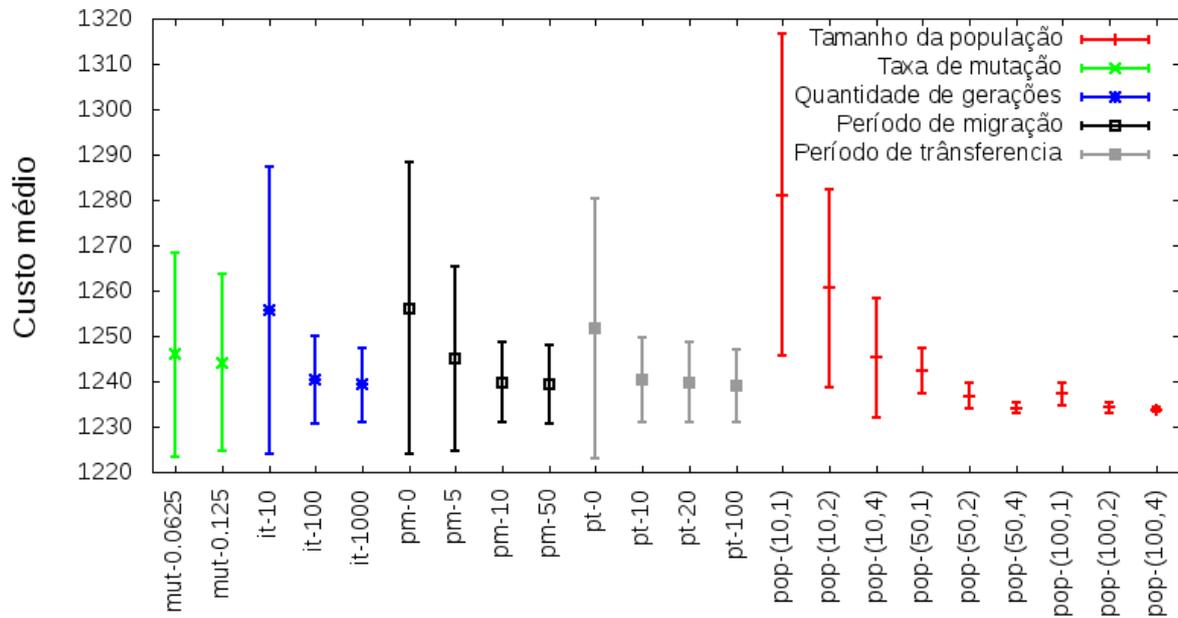


Figura 5.3: Variação de parâmetros por custo médio - Instância c03

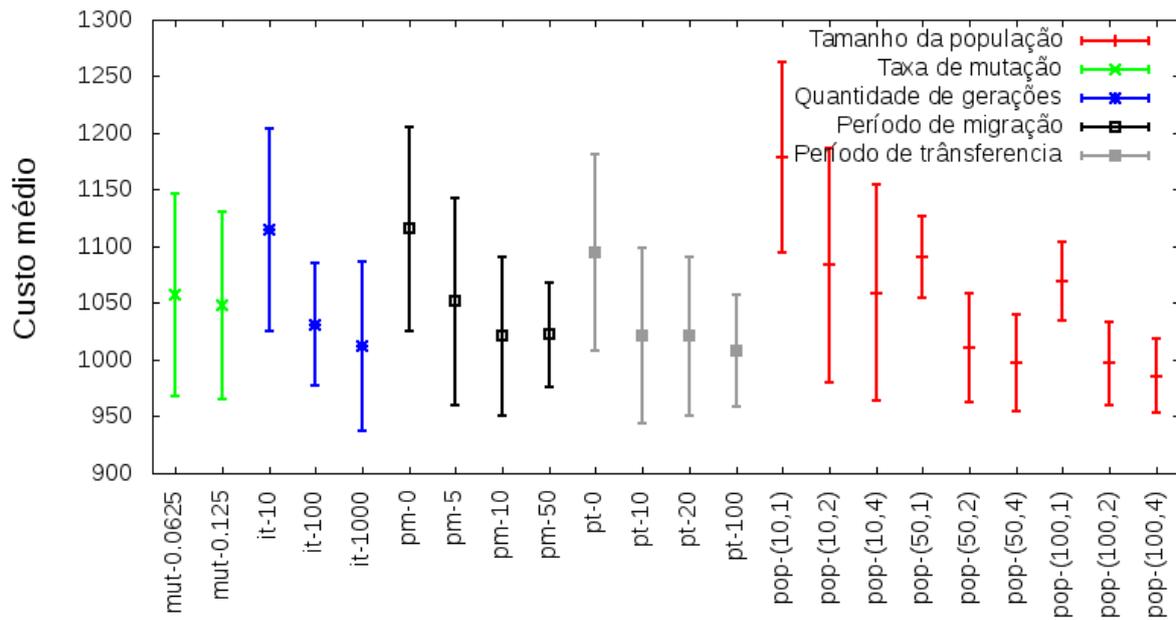


Figura 5.4: Variação de parâmetros por custo médio - Instância c04

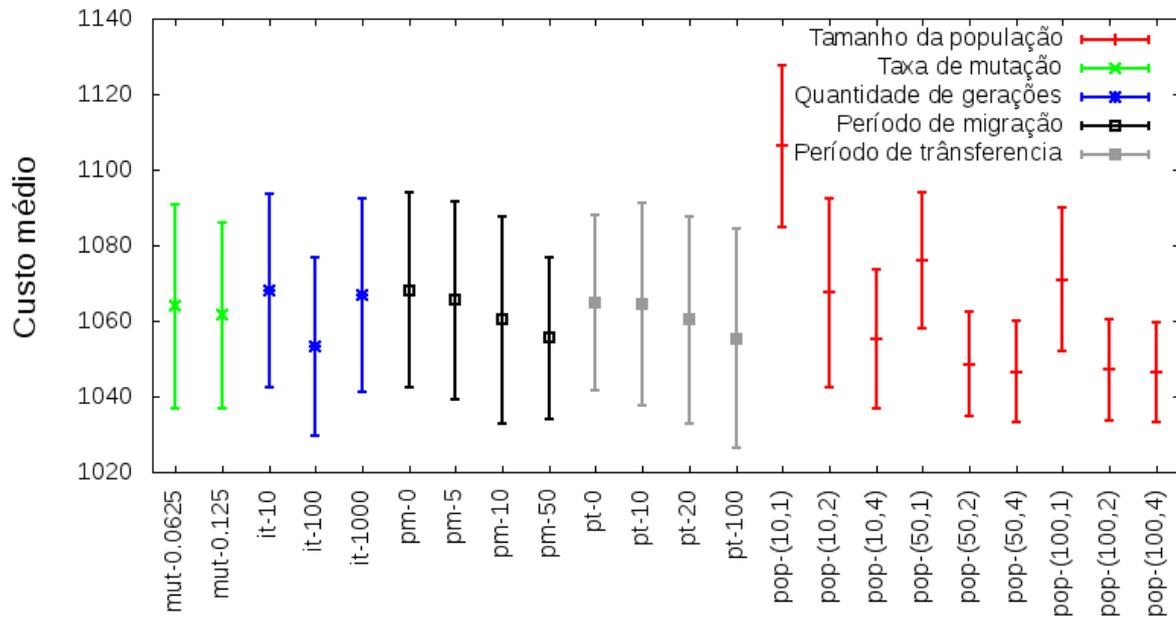


Figura 5.5: Variação de parâmetros por custo médio - Instância c05

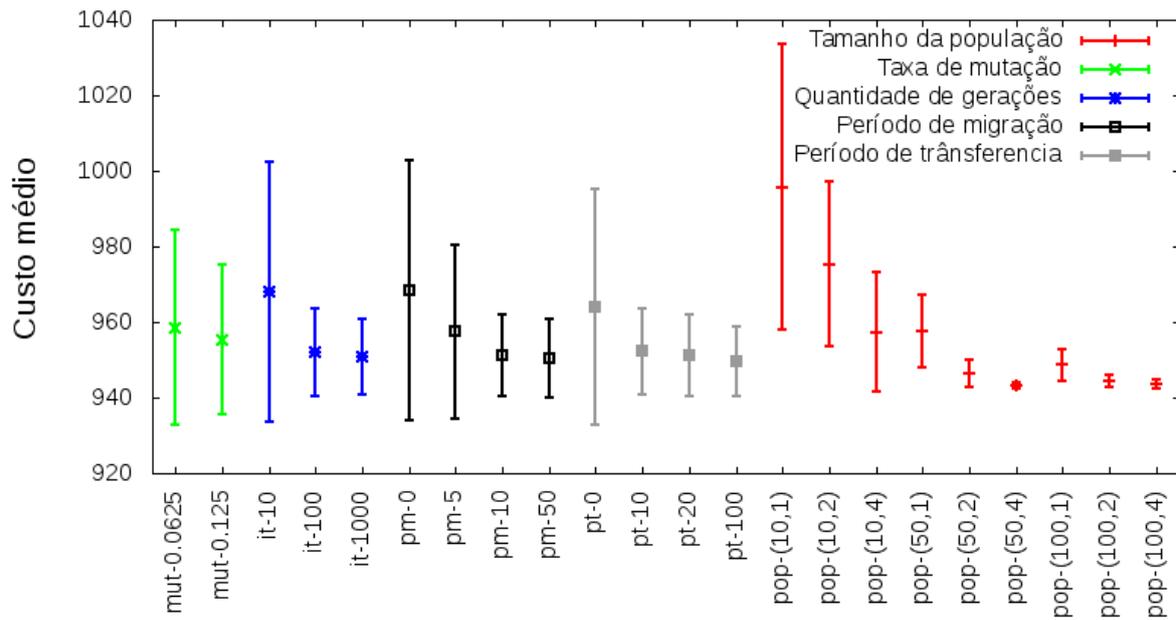


Figura 5.6: Variação de parâmetros por custo médio - Instância c06

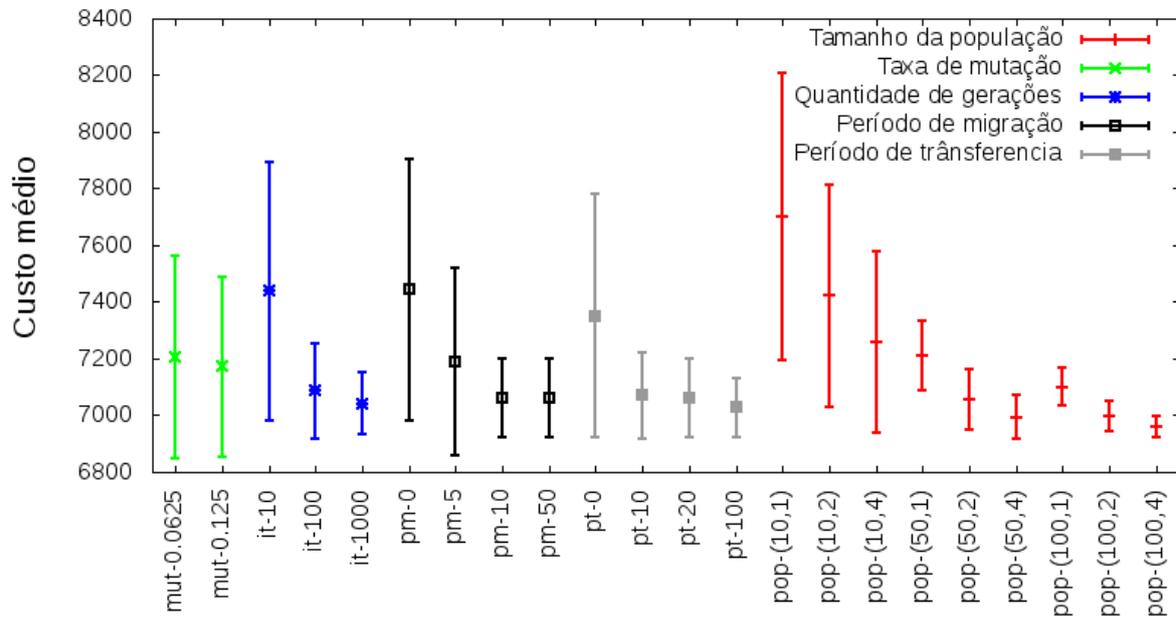


Figura 5.7: Variação de parâmetros por custo médio - Instância c07

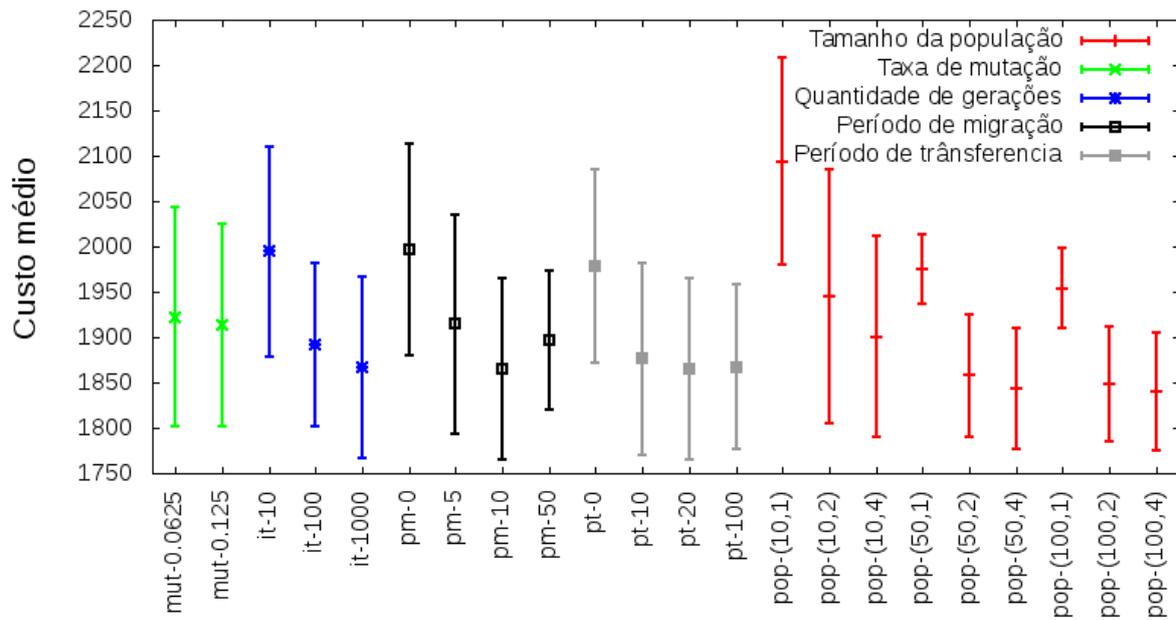


Figura 5.8: Variação de parâmetros por custo médio - Instância ch01

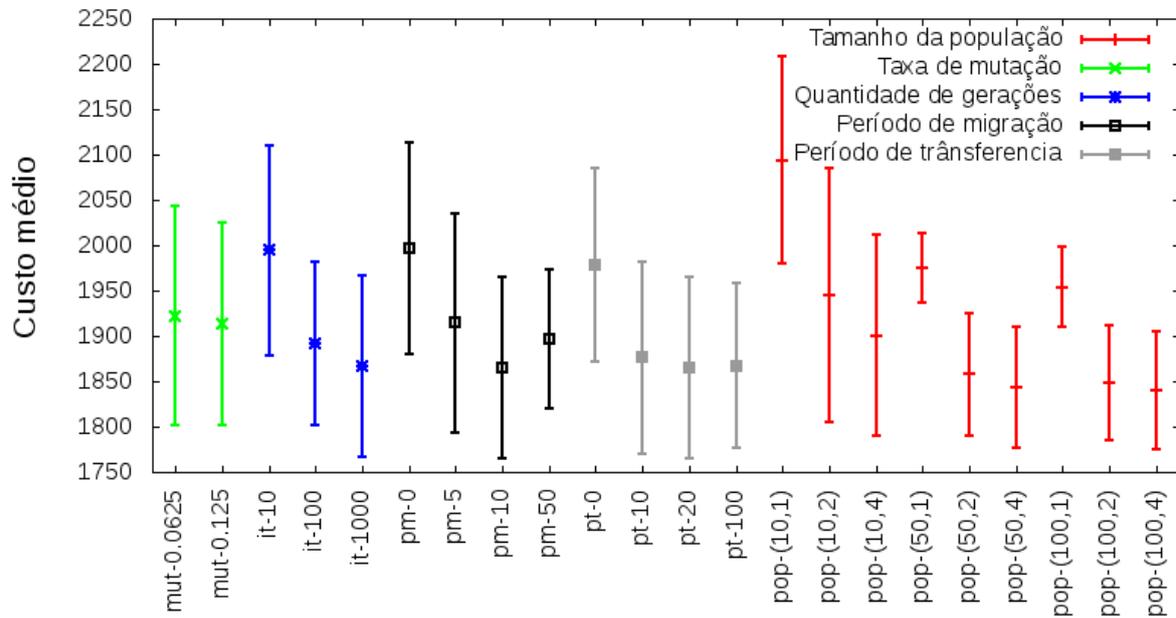


Figura 5.9: Variação de parâmetros por custo médio - Instância ch02

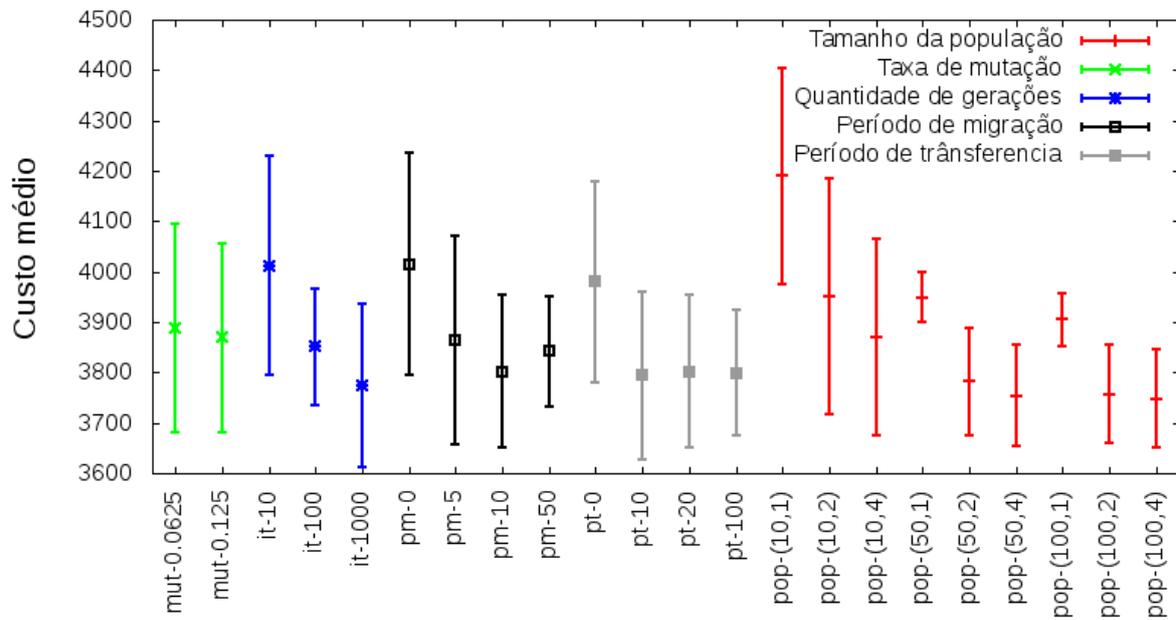


Figura 5.10: Variação de parâmetros por custo médio - Instância ch03

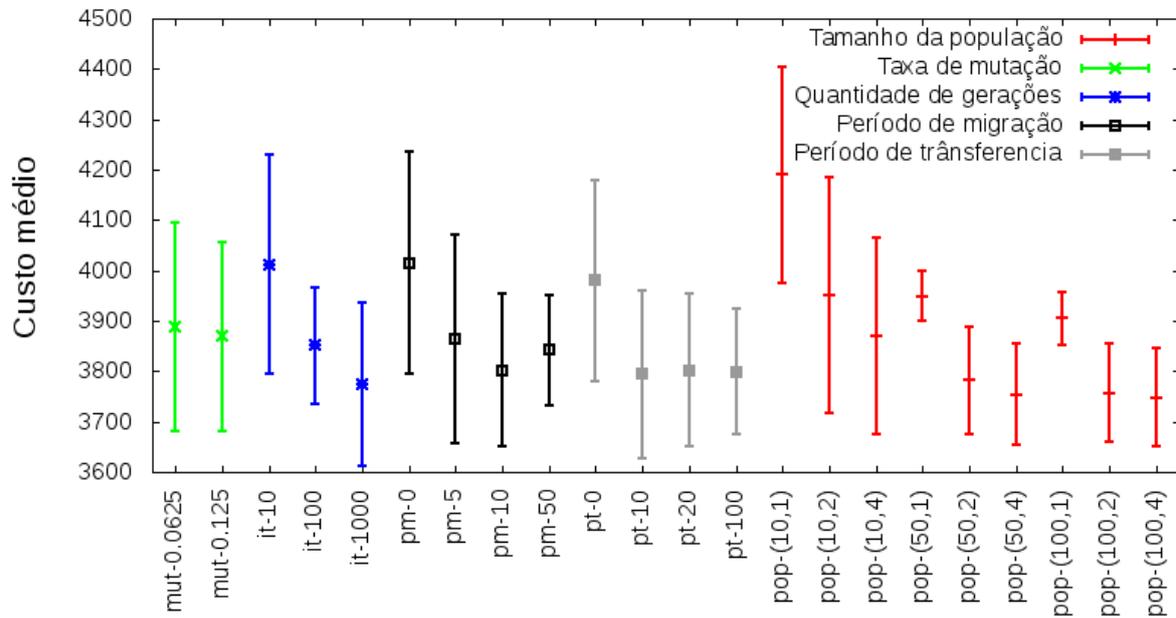


Figura 5.11: Variação de parâmetros por custo médio - Instância ch04

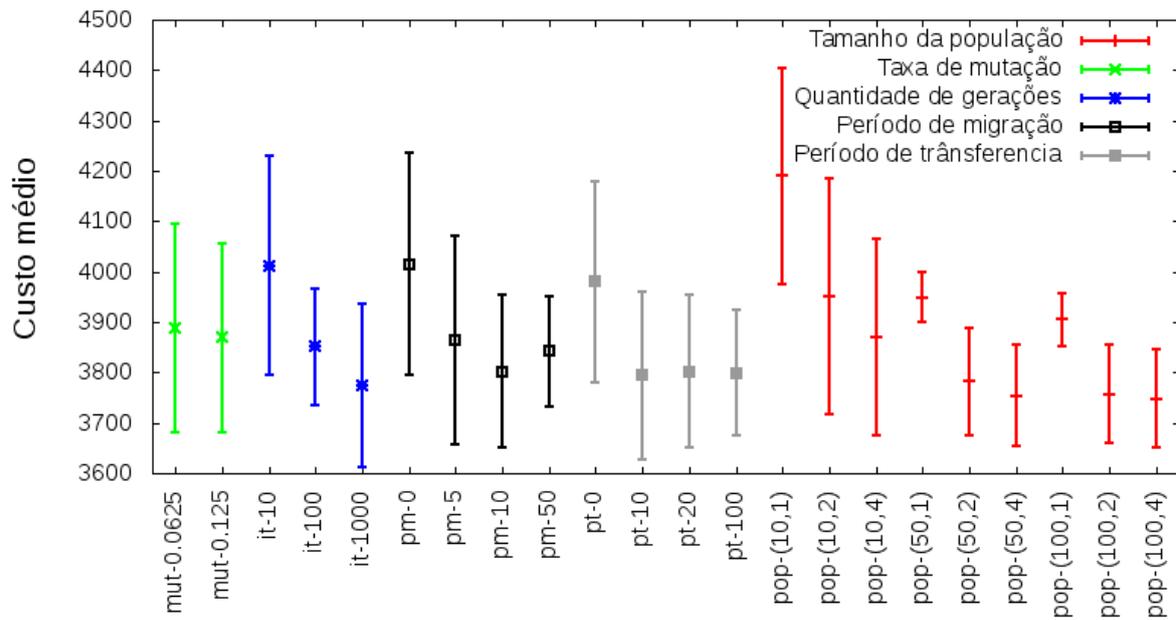


Figura 5.12: Variação de parâmetros por custo médio - Instância ch05

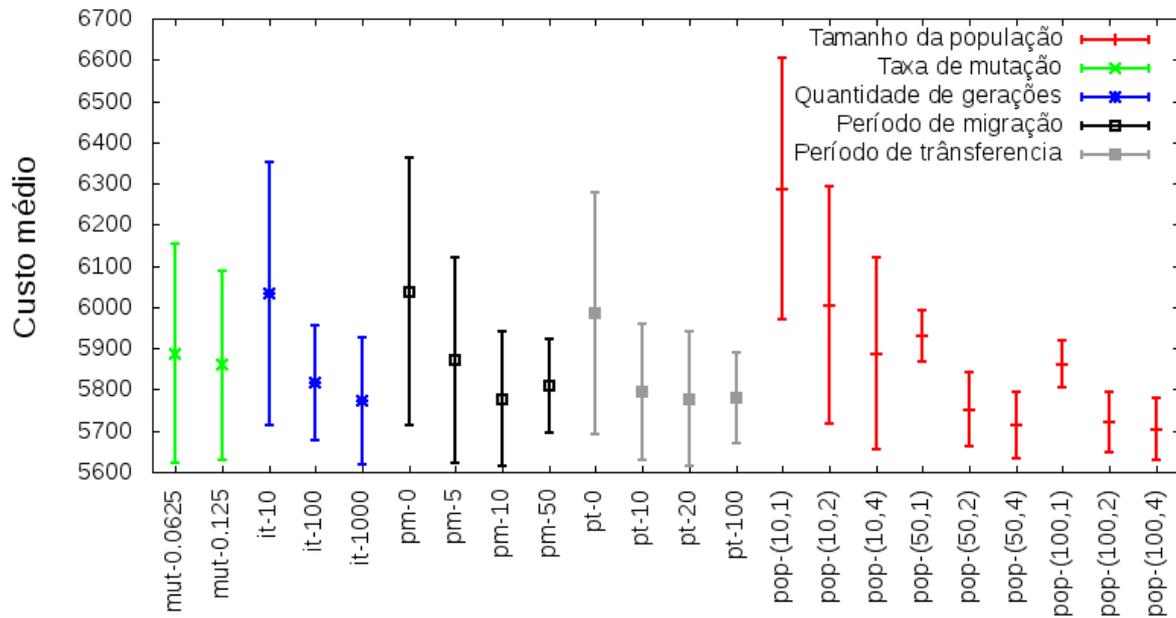


Figura 5.13: Variação de parâmetros por custo médio - Instância ch06

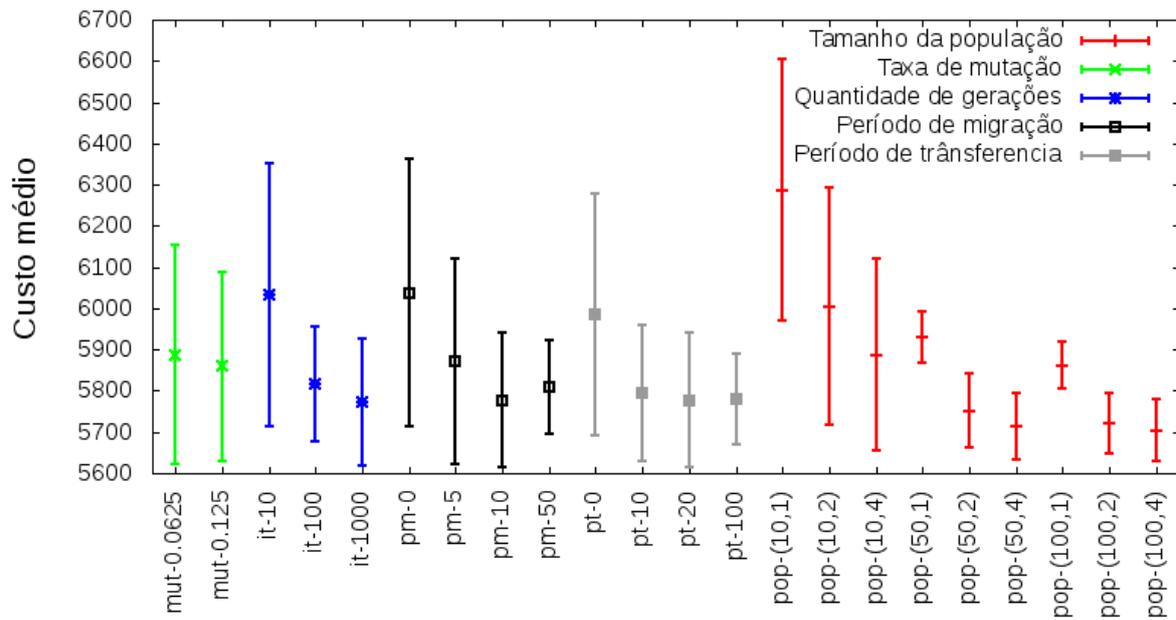


Figura 5.14: Variação de parâmetros por custo médio - Instância ch07

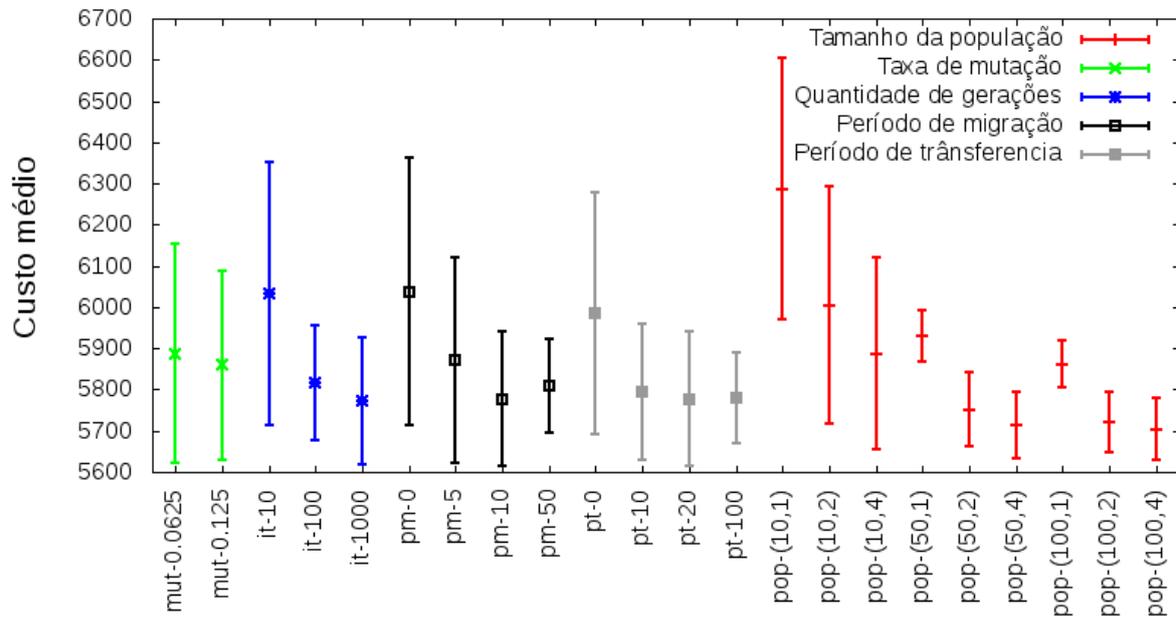


Figura 5.15: Variação de parâmetros por custo médio - Instância ch08

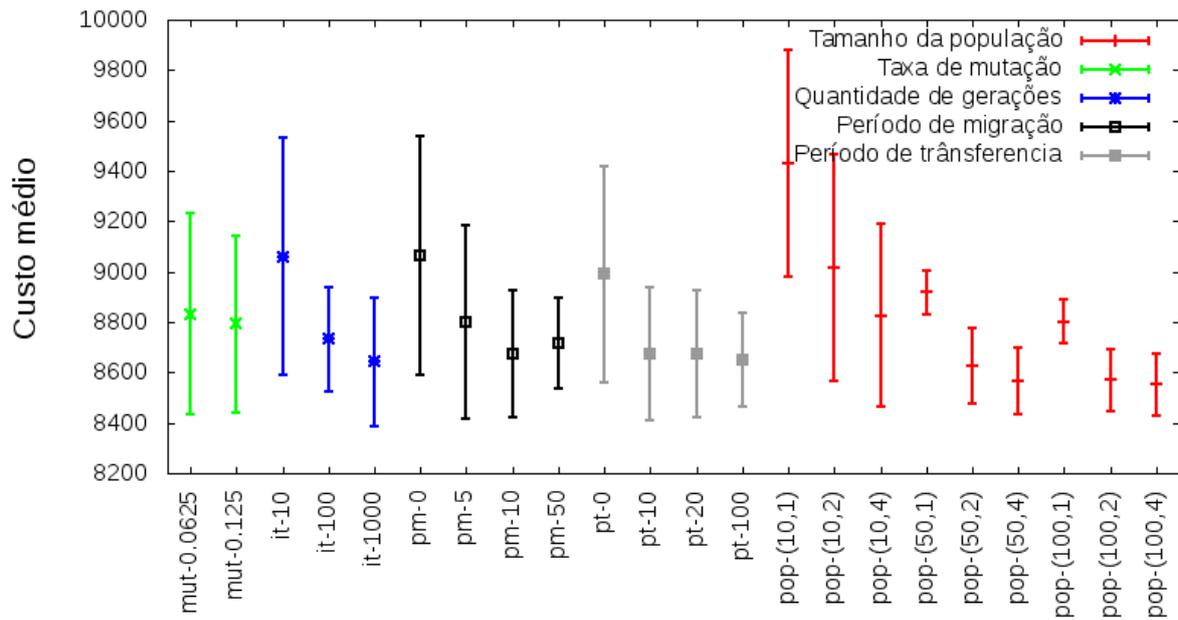


Figura 5.16: Variação de parâmetros por custo médio - Instância ch09

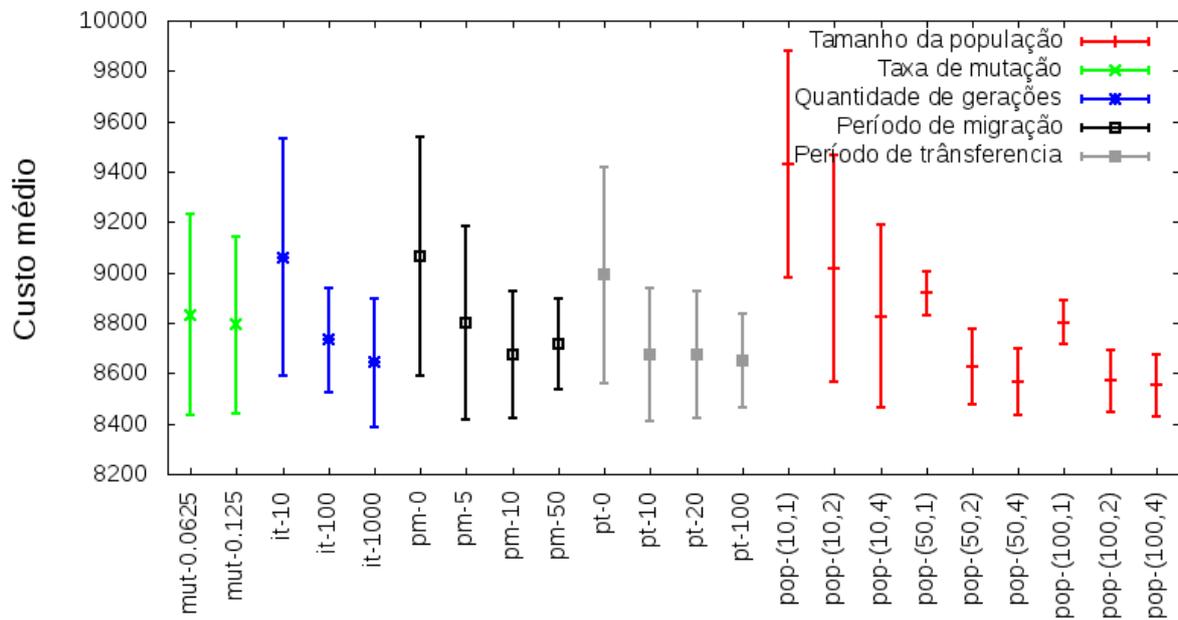


Figura 5.17: Variação de parâmetros por custo médio - Instância ch10

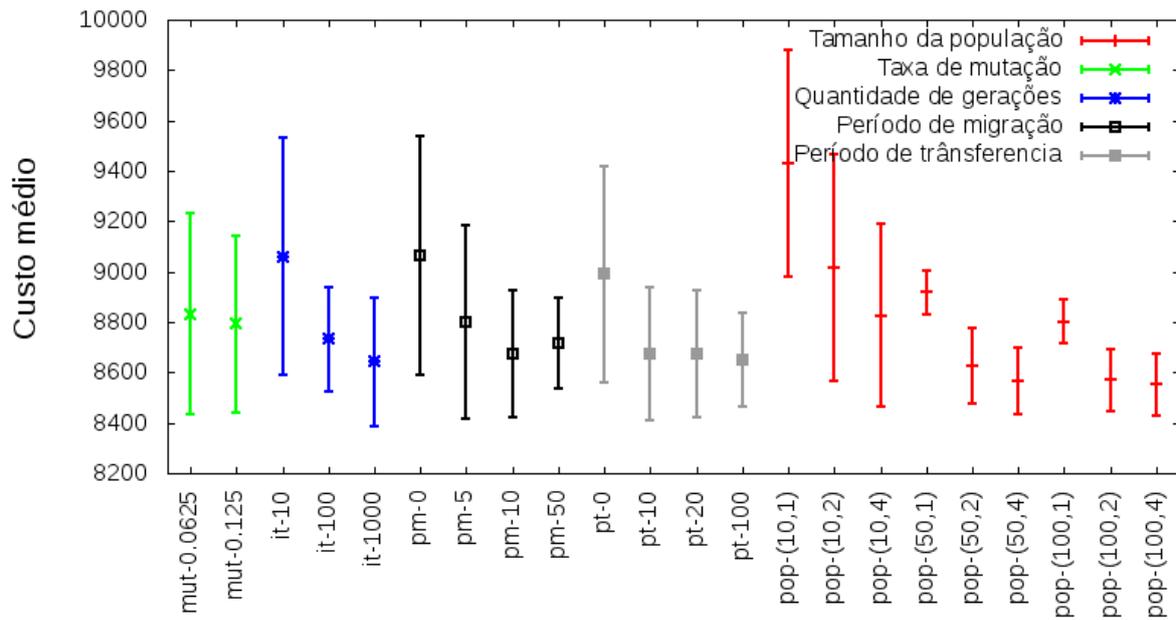


Figura 5.18: Variação de parâmetros por custo médio - Instância ch11

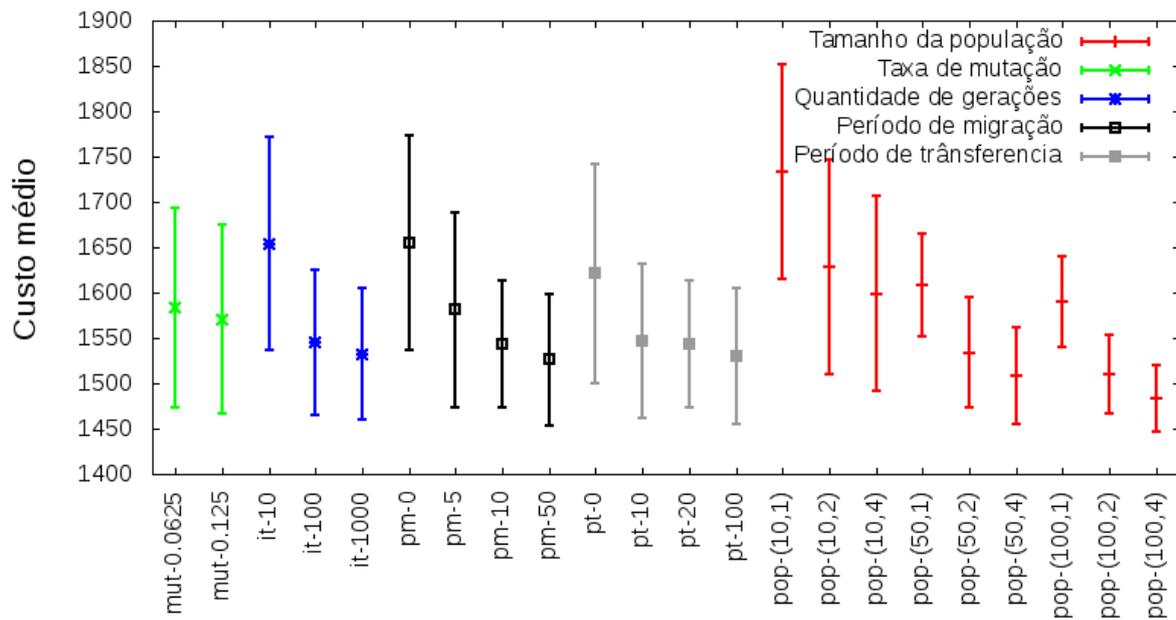


Figura 5.19: Variação de parâmetros por custo médio - Instância ch12

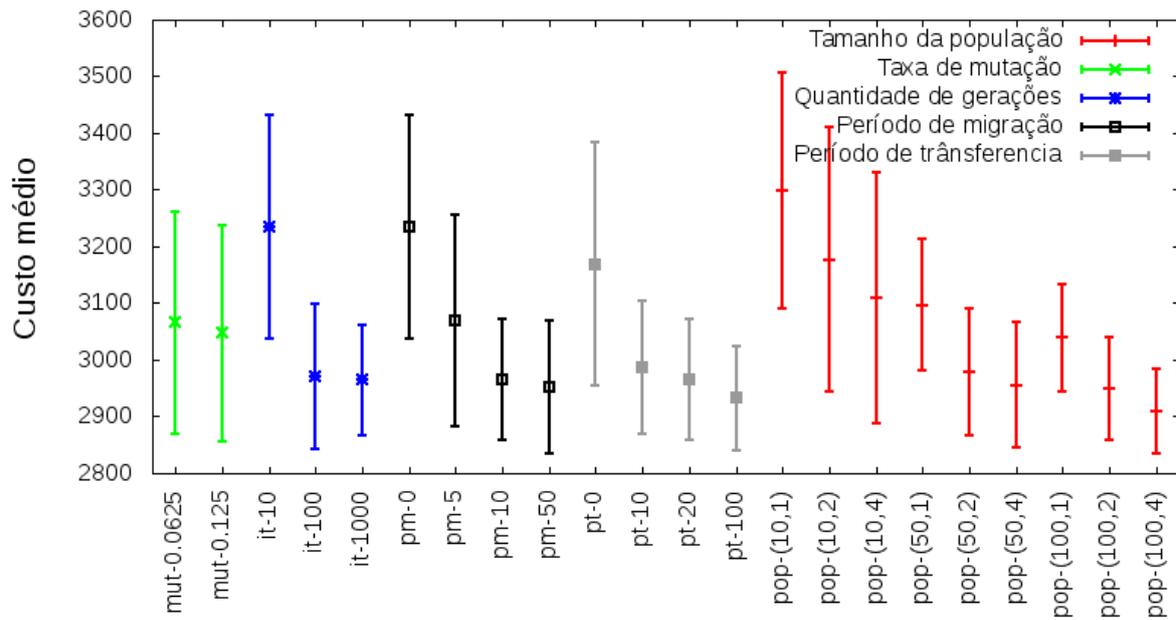


Figura 5.20: Variação de parâmetros por custo médio - Instância co01

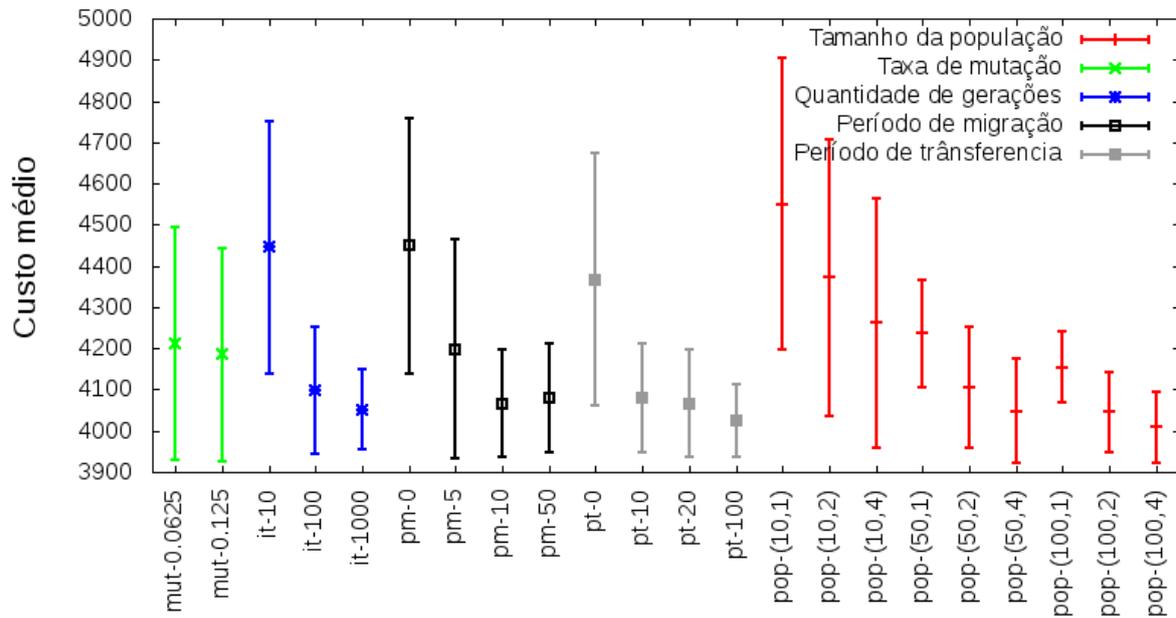


Figura 5.21: Variação de parâmetros por custo médio - Instância co02

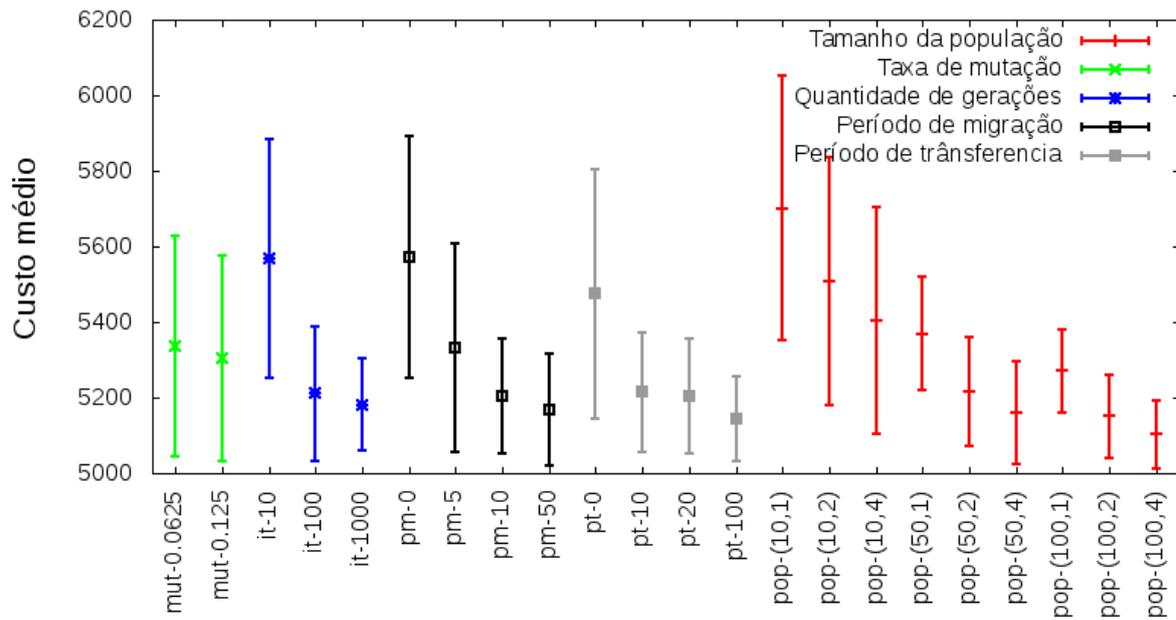


Figura 5.22: Variação de parâmetros por custo médio - Instância co03

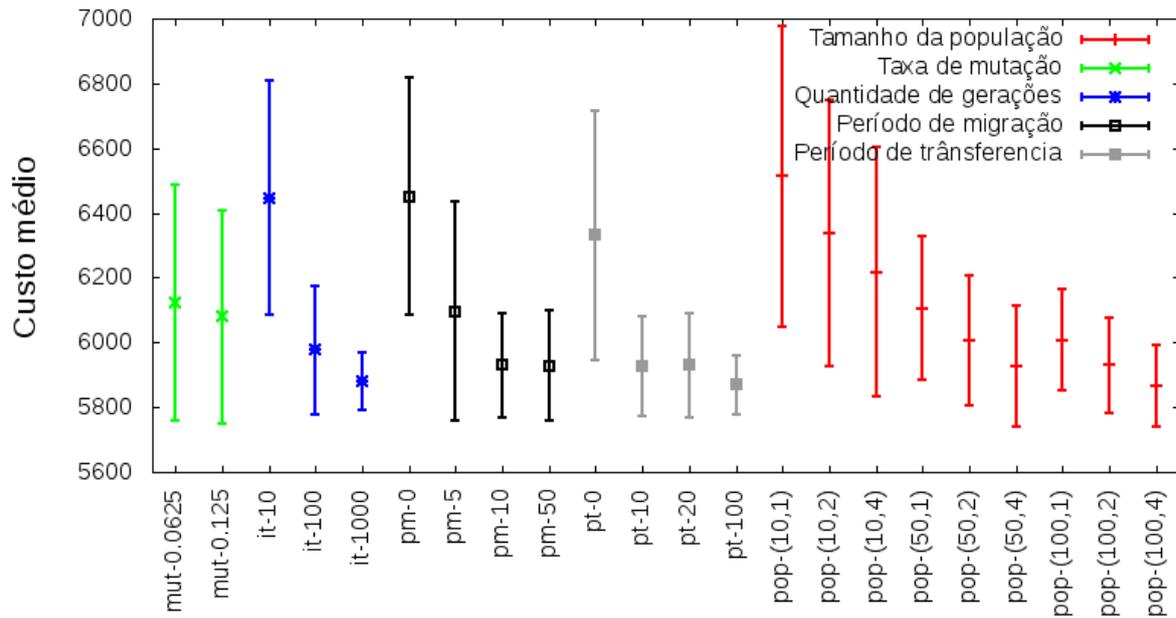


Figura 5.23: Variação de parâmetros por custo médio - Instância co04

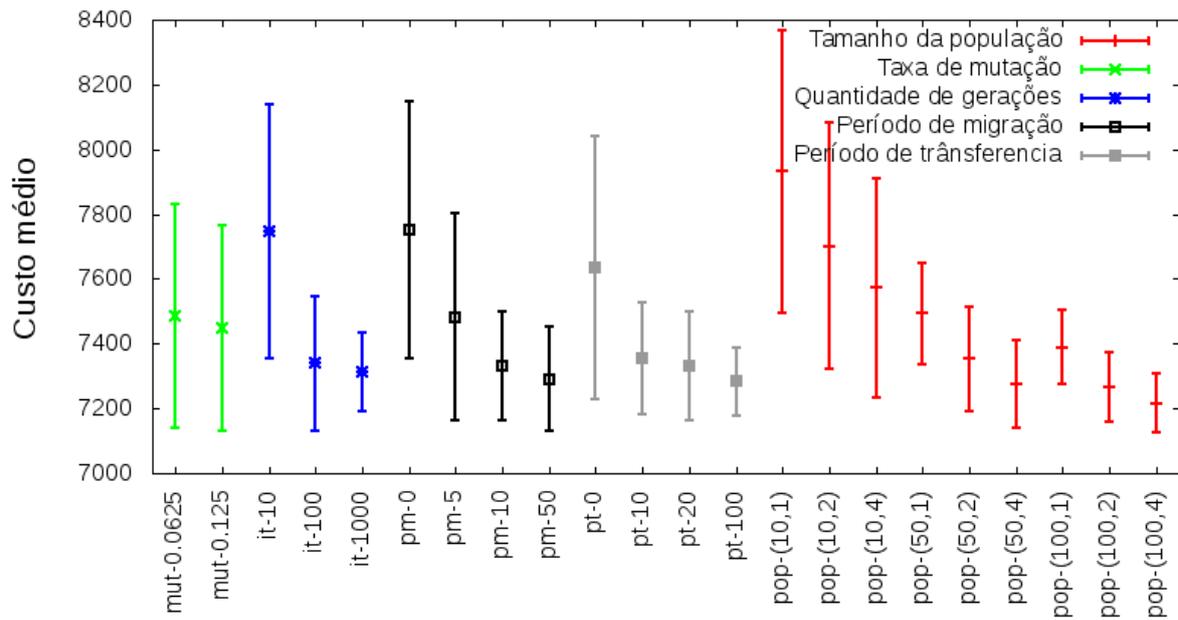


Figura 5.24: Variação de parâmetros por custo médio - Instância co05

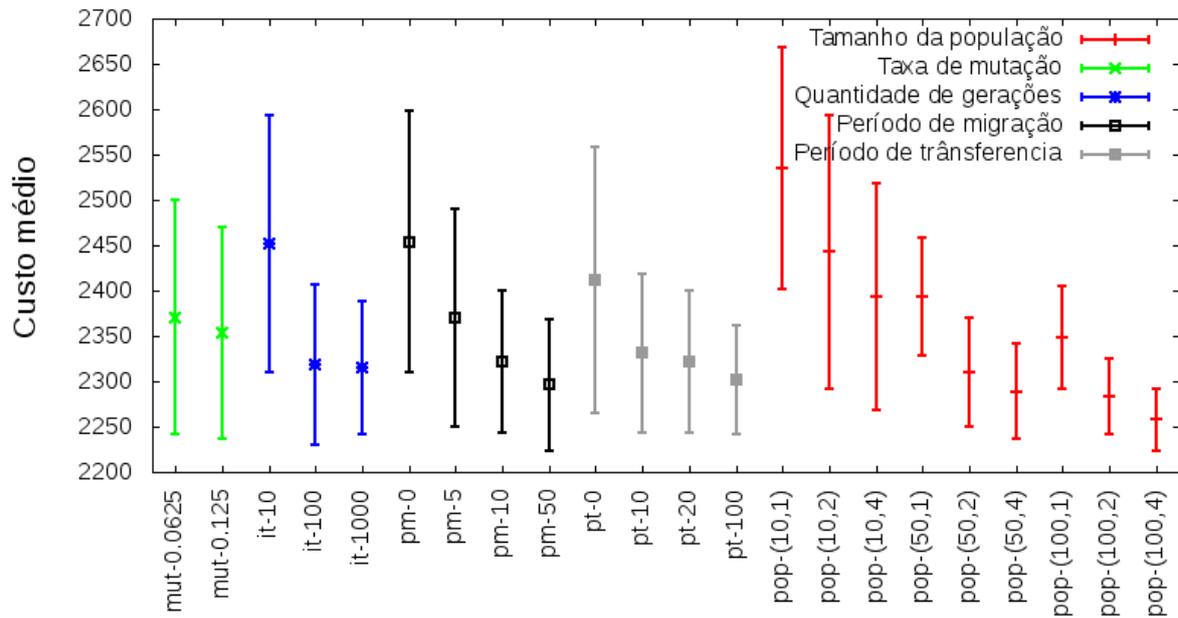


Figura 5.25: Variação de parâmetros por custo médio - Instância co06

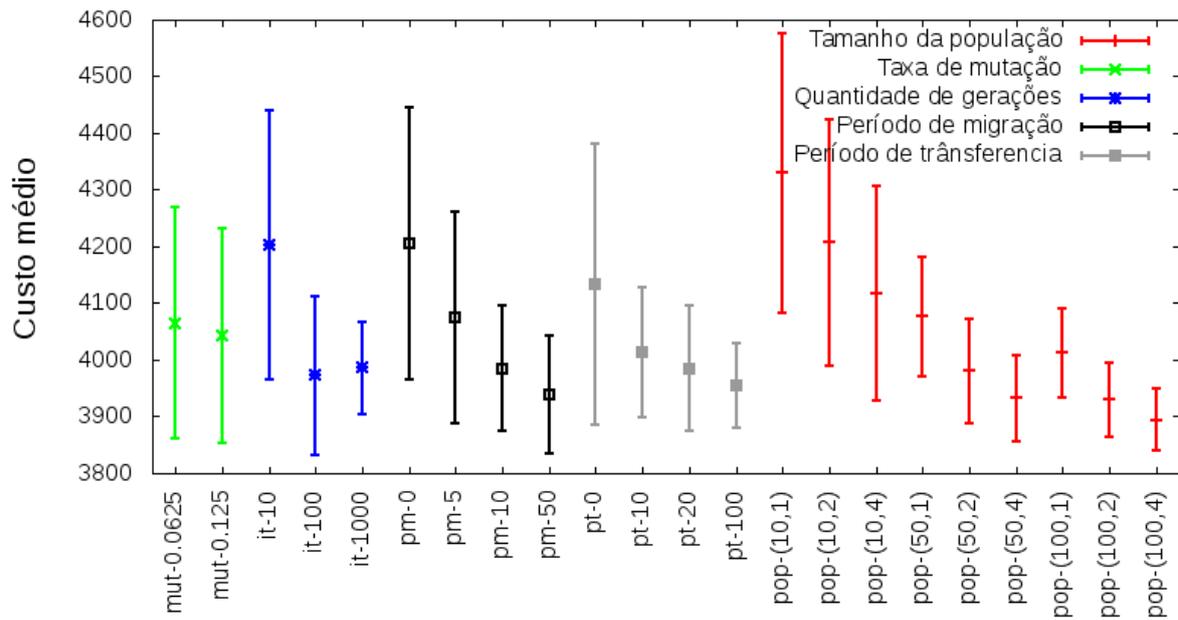


Figura 5.26: Variação de parâmetros por custo médio - Instância co07

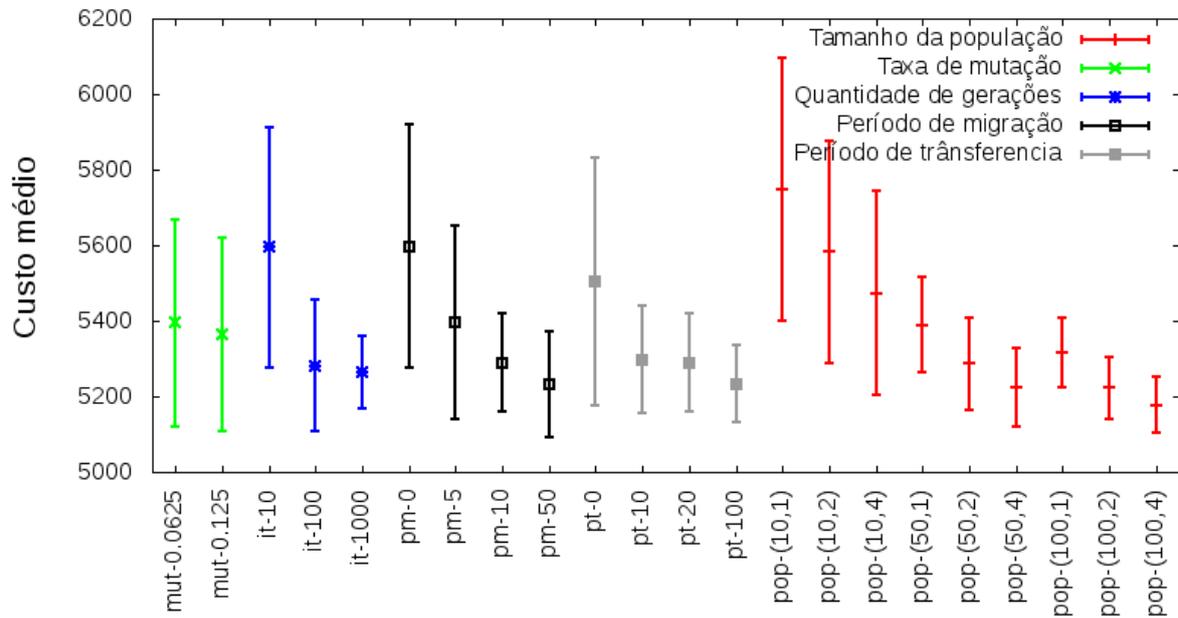


Figura 5.27: Variação de parâmetros por custo médio - Instância co08

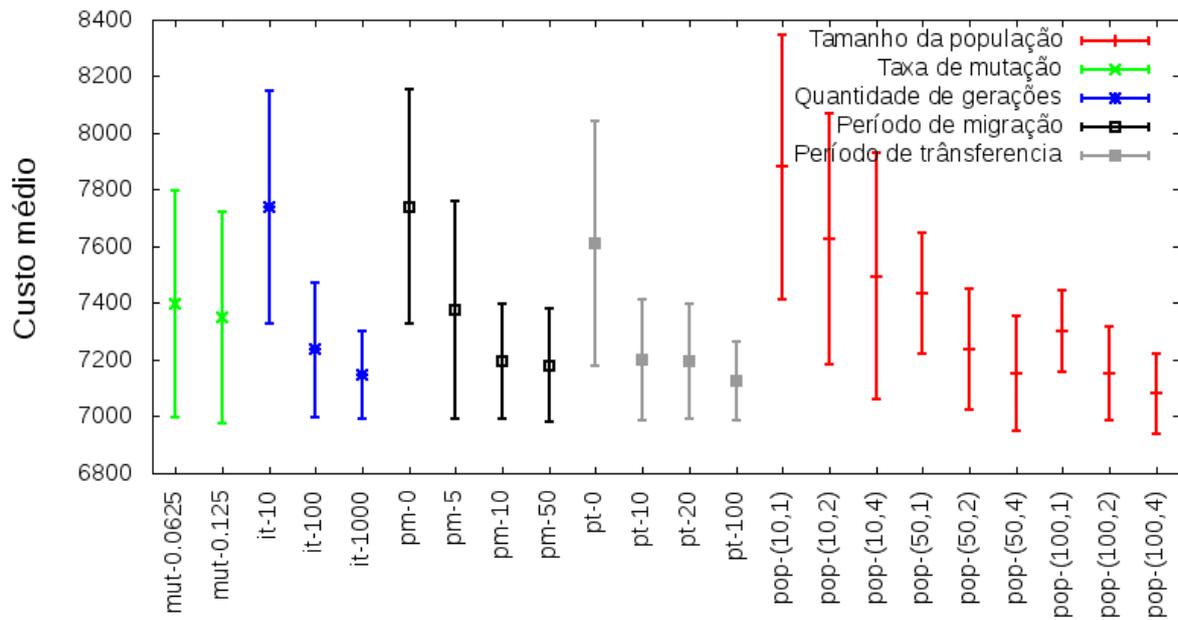


Figura 5.28: Variação de parâmetros por custo médio - Instância co09

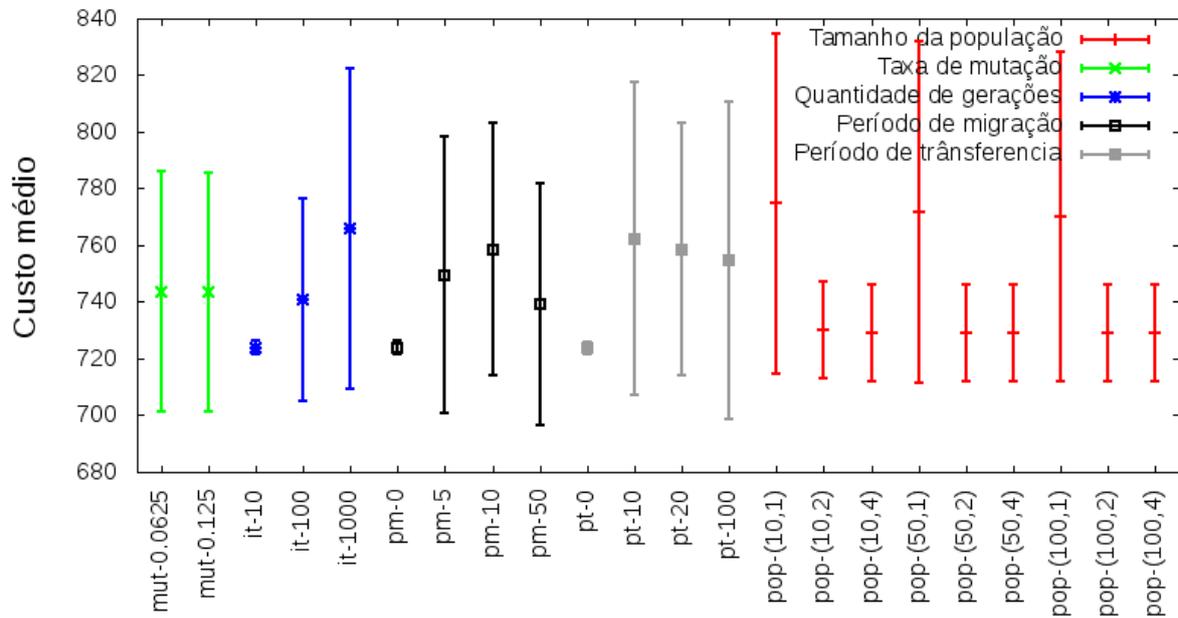


Figura 5.29: Variação de parâmetros por custo médio - Instância co10

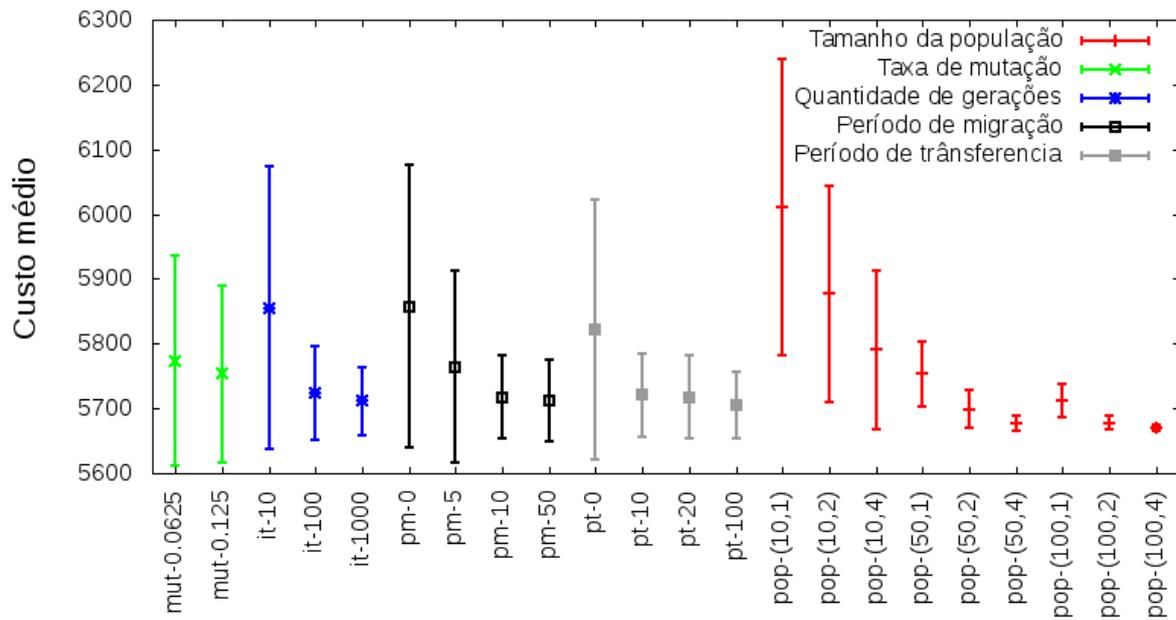


Figura 5.30: Variação de parâmetros por custo médio - Instância g01

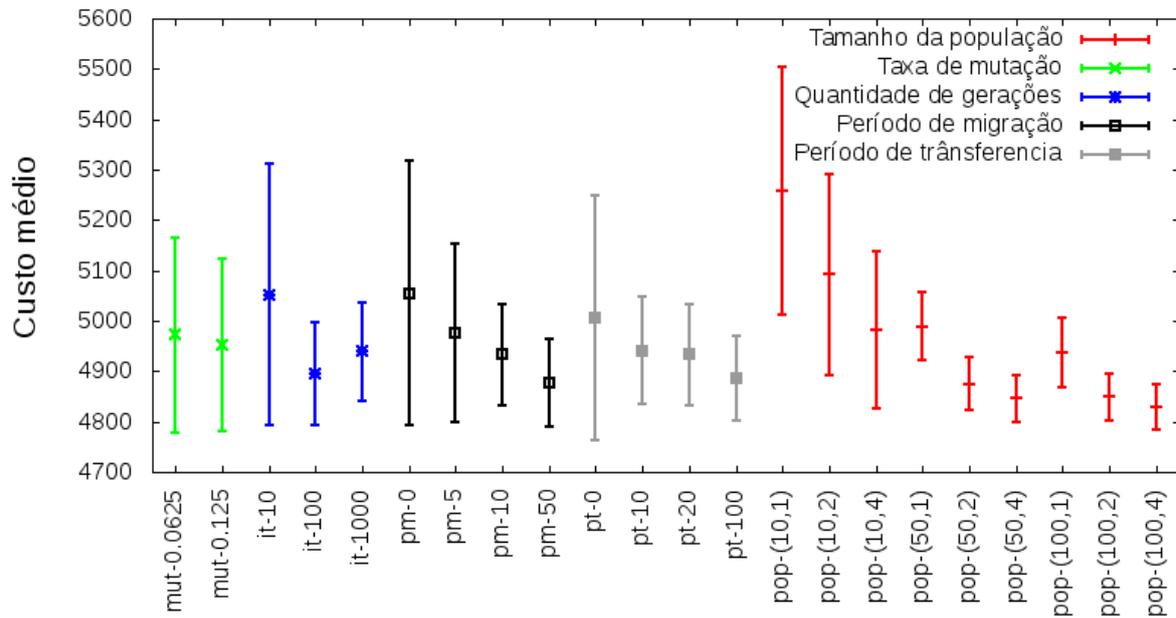


Figura 5.31: Variação de parâmetros por custo médio - Instância g02

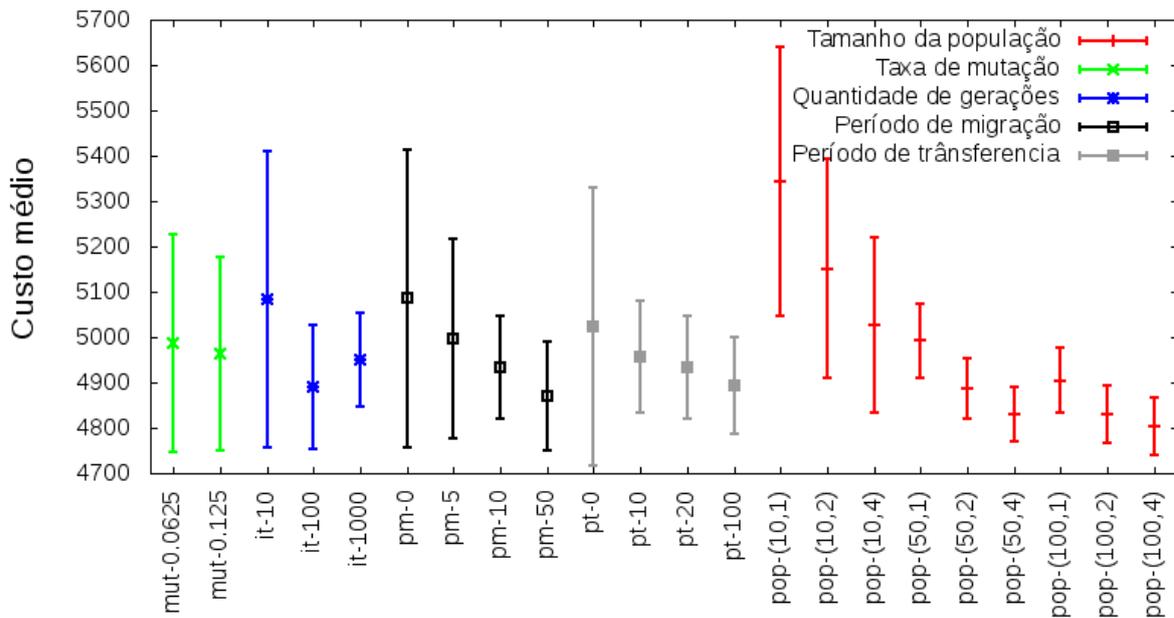


Figura 5.32: Variação de parâmetros por custo médio - Instância g03

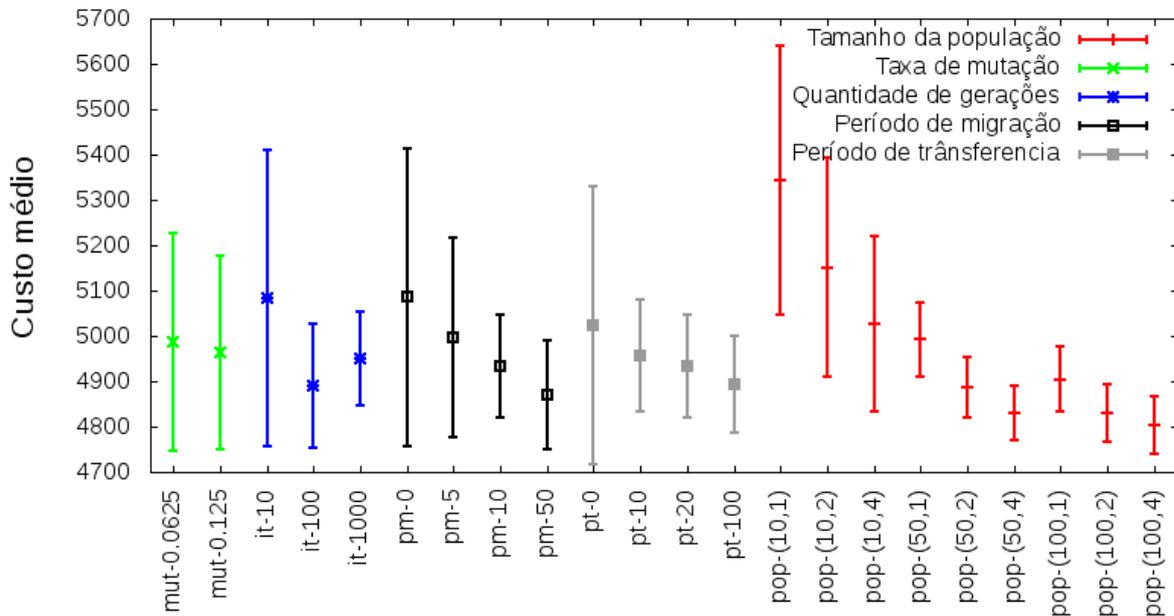


Figura 5.33: Variação de parâmetros por custo médio - Instância g04

5.2.1.1 Análise de variação - Taxa de Mutação

A taxa de mutação aparenta ser o parâmetro que causa menos impacto sozinho. A variação tanto do custo médio como do desvio padrão é muito pequena. Mesmo assim na maioria dos casos, a taxa de mutação como 0.0125 obtém tanto custo médio como desvio padrão menor.

5.2.1.2 Análise de variação - Gerações

A quantidade de gerações é um dos parâmetros mais importantes para determinar o custo computacional do método. Observamos que na maioria dos casos ocorre uma melhora significativa ao se mudar de 10 para 100 gerações. Embora exista melhora entre 100 e 1000 gerações, ela é bem menor. Acreditamos que nesse intervalo ocorra a estagnação do método na maioria dos casos, ou seja, nesse intervalo o algoritmo encontra uma boa solução e nas gerações posteriores não consegue encontrar uma solução melhor.

5.2.1.3 Análise de variação - Tamanho da população

Existem dois parâmetros que determinam o tamanho da população. O primeiro indica quantos indivíduos existem em cada população, e o segundo indica quantas populações por rota são utilizadas.

É possível identificar que picos se formam quando a quantidade de populações por rota é 1, ou seja, é utilizado um AG clássico não um AGMP. Com isso concluímos que a utilização de multi-populações melhora o resultado. Como é possível observar na configuração (10,4) onde há um total de 40 indivíduos normalmente obtém resultados melhores que a (50,1) que tem 50 indivíduos no total.

Entretanto, também pode-se observar que quando existem poucos indivíduos por população, o desvio padrão é muito alto, e que mesmo que o custo médio não reduza muito, o desvio padrão tende a reduzir ao se aumentar a quantidade de indivíduos por população.

5.2.1.4 Análise de variação - Período de Migração

Observamos que em geral obtemos melhores resultados quando realizamos migrações. Embora a diferença no custo médio seja tênue entre os diferentes períodos de migração, melhores resultados são obtidos quando o período de migrações é de 50 gerações. Além disso, nota-se que ao aumentar o período de migrações o desvio padrão diminui, ou seja, o algoritmo obtém resultados mais próximos a média.

5.2.1.5 Análise de variação - Período de Transferência

Observamos que em geral obtemos melhores resultados quando realizamos transferências. Embora a diferença seja tênue entre os diferentes períodos, melhores resultados são obtidos quando o período é de 100 gerações. Além disso, nota-se que ao aumentar o período de transferências o desvio padrão diminui, ou seja, o algoritmo obtém resultados mais próximos a média.

5.2.2 Restrições de Custo e Capacidade

Um componente da instância de um PRVMD são as restrições de custo da rota e capacidade. Ao dividir o problema em rotas, para a análise das restrições verificamos para cada rota encontrada que faz parte da solução se a mesma satisfaz ambas as restrições. Ou seja, dado o número de rotas solução de uma instância de PRVMD encontradas por nosso algoritmo, contabilizamos quantas delas atenderam a restrição de capacidade e quantas delas atenderam as restrições de custo. Então temos um valor entre 0 e 1 para cada restrição mostrando quantos por cento das rotas atenderam a essa restrição.

As tabelas 5.3, 5.4, 5.5 e 5.6, mostram para cada instância média da proporção de restrições de custo e capacidade atendidas por cada rota que faz parte da solução. Esta média leva em conta a seguinte configuração de parâmetros:

Parâmetros	Valores	Descrição
QtdIter	1000	Quantidade de iterações do método (gerações)
QtdInd	100	Quantidade de indivíduos por população
QtdPop	4	Quantidade de populações
PerMig	50	Período de migração (a cada quantas gerações)
TaxMut	0.125	Taxa de mutação
PerTrans	100	Período de transferência (a cada quantas gerações)
MinFront	3	Valor mínimo para definição da fronteira
MaxFront	7	Valor máximo para definição da fronteira

Tabela 5.2: Parâmetros dos Experimentos - Resultados finais

Esses valores foram escolhidos com base nos resultados obtidos variando todos os parâmetros e esses se mostraram, como apresentado nos gráficos das seções anteriores, na maioria das situações, os melhores valores.

Instância	Capacidade	Custo
ch01	0.900	1
ch02	0.900	0.655
ch03	0.900	0.495
ch04	0.850	1
ch05	0.850	0.522
ch06	0.850	0.400
ch07	0.833	1
ch08	0.833	0.433
ch09	0.833	0.281
ch10	0.822	1
ch11	0.822	0.377
ch12	0.822	0.300

Tabela 5.3: Média das Restrições Atendidas-Instâncias ch01-ch12

Instância	Capacidade	Custo
co01	1	0.750
co02	0.875	0.750
co03	1	0.958
co04	0.875	0.875
co05	0.450	0.890
co06	0.791	0.908
co07	0.833	0.775
co08	0.833	1
co09	0.888	0.972
co10	0.625	0.833

Tabela 5.4: Média das Restrições Atendidas - Instâncias co01-co10

Instância	Capacidade	Custo
c01	1	1
c02	1	1
c03	1	1
c04	0.875	1
c05	1	1
c06	0.888	1
c07	0.625	1

Tabela 5.5: Média das Restrições Atendidas - Instâncias c01-c07

Instância	Capacidade	Custo
g01	0.928	0.821
g02	0.916	1
g03	0.968	1
g04	0.833	1

Tabela 5.6: Média das Restrições Atendidas - Instâncias g01-g04

Observamos que, utilizando essa configuração, em algumas instâncias, todas as restrições são satisfeitas. Nos casos onde nem todas são satisfeitas, na maioria dos casos aproximadamente 80% das restrições são satisfeitas. Acreditamos que para melhorar esses resultados a atribuição

inicial dos clientes as rotas deve ser melhorada, pois esse é o ponto de partida para o algoritmo de otimização. Também pensamos que melhorando o critério para se realizar transferências podemos fazê-las de forma a melhorar as soluções.

5.2.3 Custo Médio e Melhor Custo das Soluções

Cada solução do PRVMD tem um custo, ele é dado pela soma do custo de todas as rotas que fazem parte da solução. As tabelas 5.7, 5.8, 5.9 e 5.10, mostram o custo médio e o melhor custo obtido em cada instância de cada autor. Esta média é sobre todas os testes rodados com a variações de parâmetros contidos na tabela 5.2.

Instância	Custo Médio	Melhor custo
ch01	1795.368	1768.503
ch02	1795.368	1768.503
ch03	1795.368	1768.503
ch04	3704.345	3638.926
ch05	3704.345	3638.926
ch06	3704.345	3638.926
ch07	5688.193	5677.197
ch08	5688.193	5677.197
ch09	5688.193	5677.197
ch10	8499.214	8416.301
ch11	8499.214	8416.301
ch12	8499.214	8416.301

Tabela 5.7: Custo por Instância- Instâncias ch01-ch12

Instância	Custo Médio	Melhor custo
co01	1461.543	1444.538
co02	2857.410	2819.029
co03	3940.871	3906.600
co04	5027.712	4994.396
co05	5765.408	5745.626
co06	7174.196	7118.177
co07	2242.582	2209.164
co08	3877.708	3826.785
co09	5131.142	5060.878
co10	6967.518	6921.669

Tabela 5.8: Custo por Instância - Instâncias co01-co10

Instância	Custo Médio	Melhor custo
c01	729.458	704.039
c02	563.887	562.110
c03	782.517	756.650
c04	1233.745	1233.386
c05	966.743	962.738
c06	1037.455	1018.435
c07	944.159	943.164

Tabela 5.9: Custo por Instância - Instâncias c01-c07

Instância	Custo Médio	Melhor custo
g01	6929.605	6917.055
g02	5666.766	5665.972
g03	4811.655	4791.447
g04	4816.743	4734.824

Tabela 5.10: Custo Médio por Instância - Instâncias g01-g04

Capítulo 6

Conclusão

Este trabalho apresentou uma abordagem meta-heurística para o PRVMD. Este é um problema desafiador, principalmente por apresentar um espaço de busca muito grande. A abordagem apresentada decompõe o PRVMD em subproblemas que cooperam para solucionar o problema original. Cada subproblema tenta encontrar a melhor solução para um determinado veículo.

A otimização da rota para cada veículo é feita de forma independente no domínio de seu subproblema, sem interferir nas outras rotas, utilizando o AG.

Todavia, como mostraram os resultados dos experimentos, em alguns casos essa abordagem não conseguiu atender a todas as restrições do PRVMD. Mesmo assim, acreditamos que ela não está longe de conseguir cumprir esse objetivo.

Uma alternativa a considerar seria utilizar outros algoritmos de otimização para encontrar a melhor rota.

Além disso, acreditamos que o ponto chave para as restrições serem sempre atendidas está na atribuição de clientes a rotas. Diversos algoritmos poderiam ser utilizados para realizar a atribuição inicial, como por exemplo algoritmos que utilizam densidade ao invés de distância. Ainda, para decidir quais serão as transferências de clientes a serem realizadas, mais uma vez abordagens baseadas em densidade poderiam ser utilizadas.

Referências Bibliográficas

- Chao, I.-M., Golden, B. L. e Wasil, E. (1993). A new heuristic for the multi-depot vehicle routing problem that improves upon best-known solutions. *American Journal of Mathematical and Management Sciences*, 13(3-4):371–406.
- Christofides, N. e Eilon, S. (1969). An algorithm for the vehicle-dispatching problem. *OR*, 20(3):309–318.
- Coley, D. A. (1999). *AN INTRODUCTION TO GENETIC ALGORITHMS FOR SCIENTISTS AND ENGINEERS*. World Scientific Publishing Co. Pte. Ltd.
- Cordeau, J., Gendreau, M. e Laporte, G. (1997). A tabu search heuristic for periodic and multi-depot vehicle routing problems. *Networks*, 30(2):105–119.
- Dorigo, M. e Gambardella, L. M. (1997). Ant colony system: a cooperative learning approach to the traveling salesman problem. *IEEE Transactions on evolutionary computation*, 1(1):53–66.
- Engelbrecht, A. P. (2007). *Computational Intelligence: An Introduction*. Wiley Publishing, 2nd edition.
- Gillett, B. E. e Johnson, J. G. (1976). Multi-terminal vehicle-dispatch algorithm. *Omega*, 4(6):711 – 718.
- Holland, J. (1975). *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. University of Michigan Press.
- Jozefowicz, N., Semet, F. e Talbi, E. (2009). An evolutionary algorithm for the vehicle routing problem with route balancing. *Elsevier*, 195:761–769.
- Neto, A. J. S., Becceneri, J., Biondi, L., Silva, J. D. S. e Luz, E. F. P. (2009). *Técnicas de Inteligência Computacional Inspiradas na Natureza: Aplicações em Problemas Inversos em Transferência Radiativa*, páginas 35–42. SBMAC - São Carlos - SP.
- Oliveira, F. B., Enayatifar, R., Sadaei, H. J., Guimarães, F. G. e Potvind, J. (2016). A cooperative coevolutionary algorithm for the multi-depot vehicle routing problem. *Elsevier*, 54:398–402.
- Souza, M. J. F. (2005). Inteligencia computacional para otimização. <https://www.dca.ufrn.br/~estefane/metaheuristicas/InteligenciaComputacional.pdf>. Acessado em 08/11/2017.
- Ulder, N. L. J., Aarts, E. H. L., Bandelt, H.-J., van Laarhoven, P. J. M. e Pesch, E. (1991). *Genetic local search algorithms for the traveling salesman problem*, páginas 109–116. Springer Berlin Heidelberg, Berlin, Heidelberg.

Vidal, T., Crainic, T. G., Gendreau, M. e Prins, C. (2014). Implicit depot assignments and rotations in vehicle routing heuristics. *European Journal of Operational Research*, 237(1):15–28.

Weise, T. (2011). *Global Optimization Algorithms - Theory and Application* -. 3rd edition.

Yu, X. e Gen, M. (2010). *Introduction to Evolutionary Algorithms*. Springer London.

Apêndice A

Instâncias

A seguir apresentamos as instâncias utilizadas para teste do algoritmo proposto bem como a figura da localização de cada cliente e depósito para cada instância. Em instâncias onde a duração máxima do serviço tem o valor 0 (zero) significa que não há restrição de duração, isso quer dizer que não há limite para a duração da rota.

A.1 Instâncias propostas por Christofides e Eilon (1969):

A.1.1 Instância c01

- 2 veículos;
- 50 clientes;
- 4 depósitos;
- Duração máxima do serviço: 0;
- Capacidade máxima de cada veículo: 80.

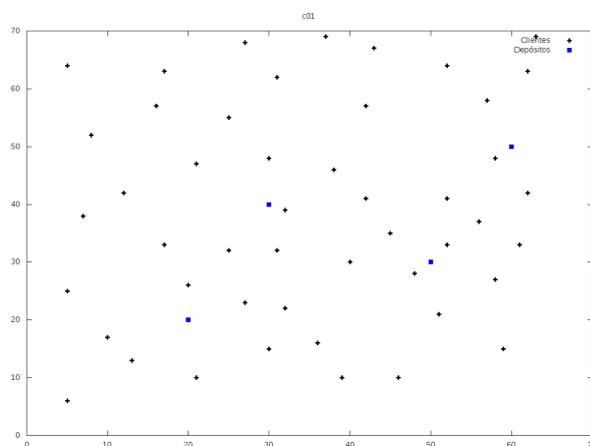


Figura A.1: Mapa - Instância c01

A.1.2 Instância c02

- 2 veículos;
- 50 clientes;
- 4 depósitos;
- Duração máxima do serviço: 0;
- Capacidade máxima de cada veículo: 160.

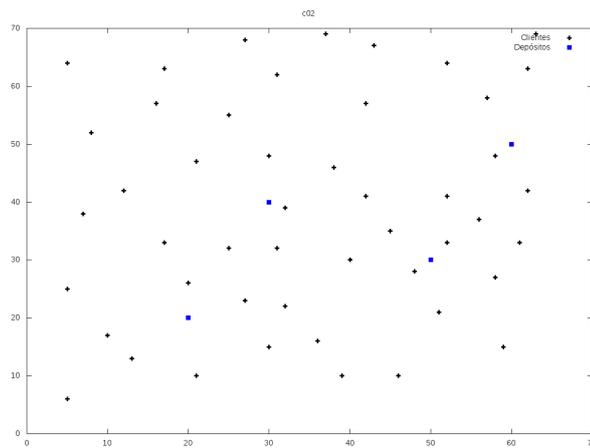


Figura A.2: Mapa - Instância c02

A.1.3 Instância c03

- 3 veículos;
- 75 clientes;
- 5 depósitos;
- Duração máxima do serviço: 0;
- Capacidade máxima de cada veículo: 140.

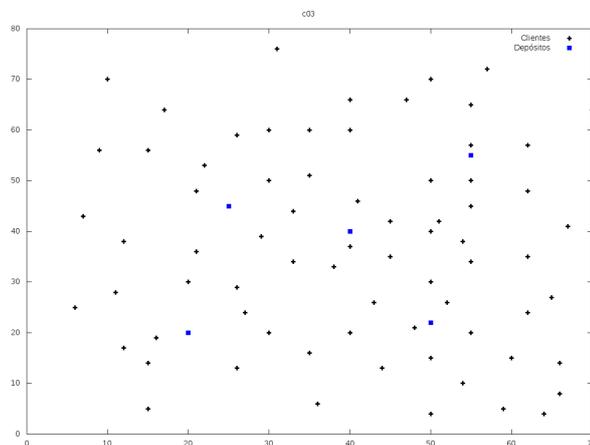


Figura A.3: Mapa - Instância c03

A.1.4 Instância c04

- 8 veículos;
- 100 clientes;
- 2 depósitos;
- Duração máxima do serviço: 0;
- Capacidade máxima de cada veículo: 100.

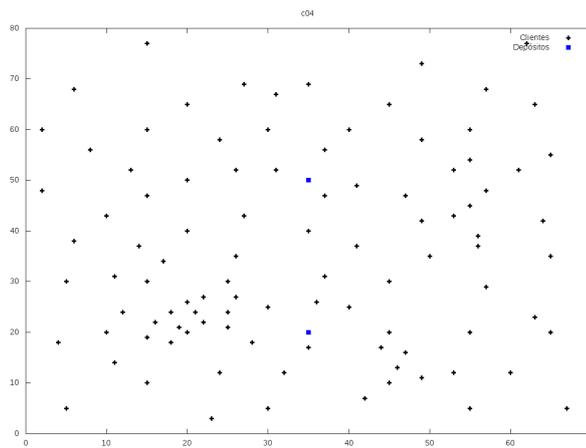


Figura A.4: Mapa - Instância c04

A.1.5 Instância c05

- 5 veículos;
- 100 clientes;
- 2 depósitos;
- Duração máxima do serviço: 0;
- Capacidade máxima de cada veículo: 200.

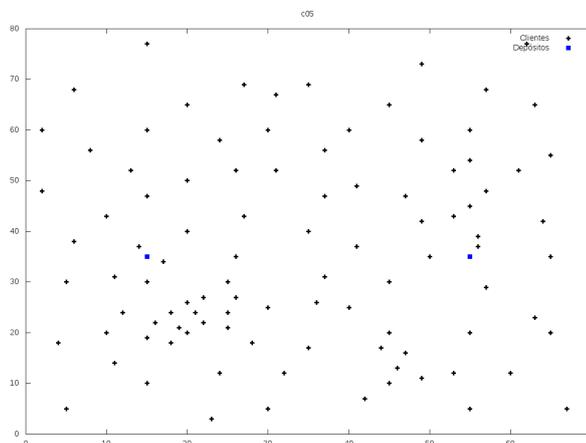


Figura A.5: Mapa - Instância c05

A.1.6 Instância c06

- 6 veículos;
- 100 clientes;
- 3 depósitos;
- Duração máxima do serviço: 0;
- Capacidade máxima de cada veículo: 100.

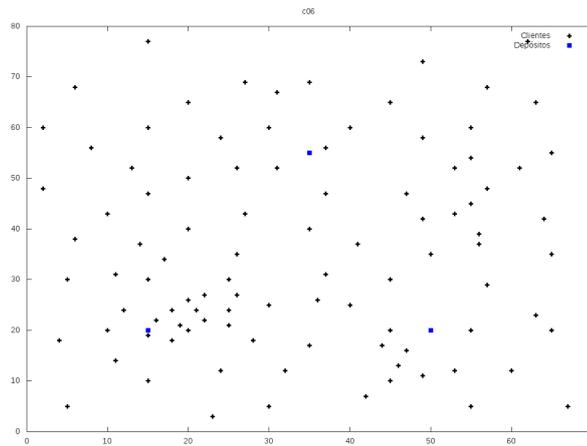


Figura A.6: Mapa - Instância c06

A.1.7 Instância c07

- 4 veículos;
- 100 clientes;
- 4 depósitos;
- Duração máxima do serviço: 0;
- Capacidade máxima de cada veículo: 100.

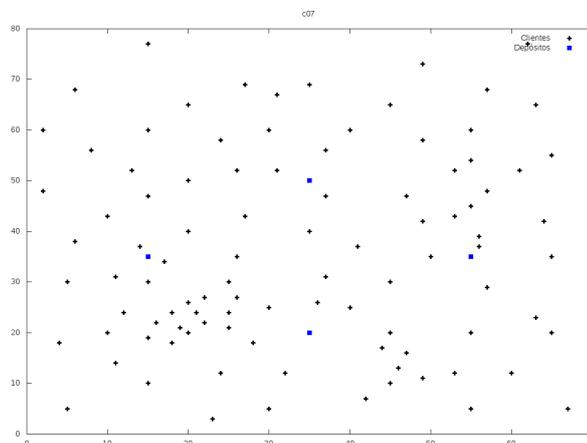


Figura A.7: Mapa - Instância c07

A.2 Instâncias propostas por Gillett e Johnson (1976):

A.2.1 Instância g01

- 12 veículos;
- 249 clientes;
- 2 depósitos;
- Duração máxima do serviço: 310;
- Capacidade máxima de cada veículo: 500.

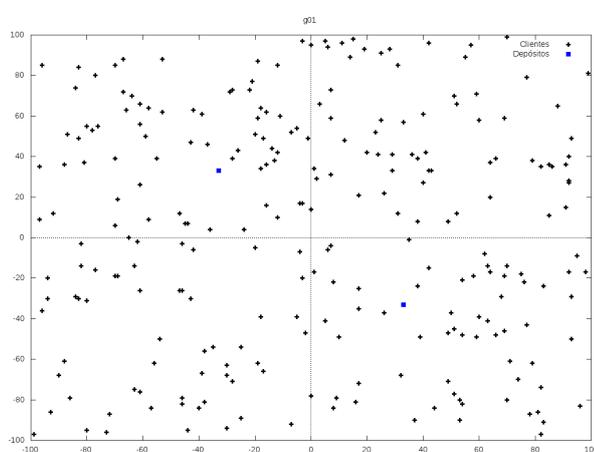


Figura A.8: Mapa - Instância g01

A.2.2 Instância g02

- 12 veículos;
- 249 clientes;
- 3 depósitos;
- Duração máxima do serviço: 310;
- Capacidade máxima de cada veículo: 500.

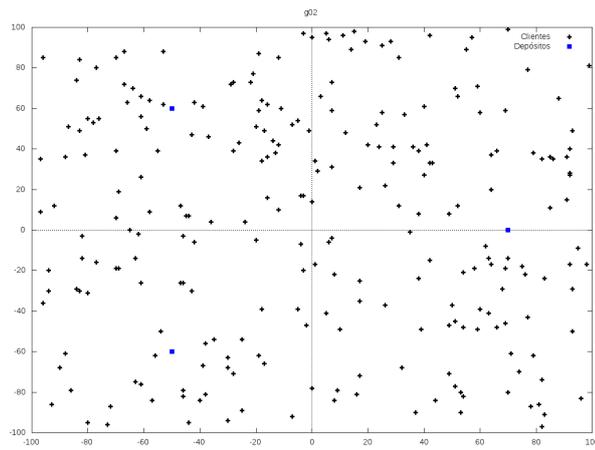


Figura A.9: Mapa - Instância g02

A.2.3 Instância g03

- 8 veículos;
- 249 clientes;
- 4 depósitos;
- Duração máxima do serviço: 310;
- Capacidade máxima de cada veículo: 500.

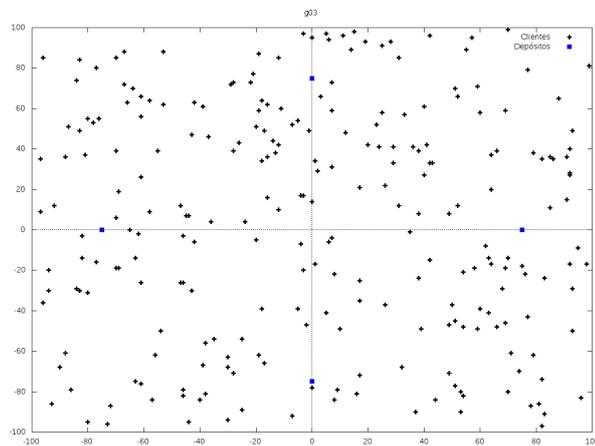


Figura A.10: Mapa - Instância g03

A.2.4 Instância g04

- 6 veículos;
- 249 clientes;
- 5 depósitos;
- Duração máxima do serviço: 310;

- Capacidade máxima de cada veículo: 500.

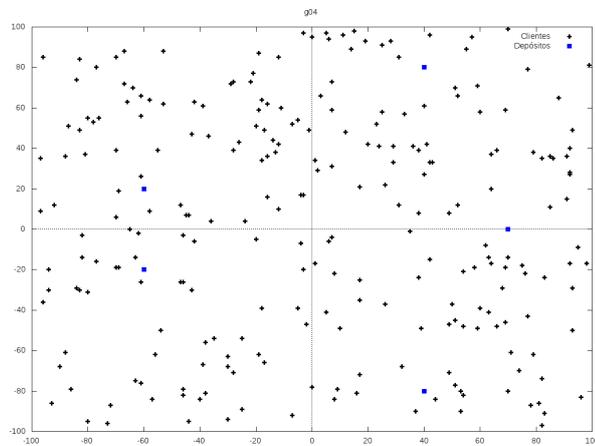


Figura A.11: Mapa - Instância g04

A.3 Instâncias propostas por Chao et al. (1993):

A.3.1 Instância ch01

- 5 veículos;
- 80 clientes;
- 2 depósitos;
- Duração máxima do serviço: 0;
- Capacidade máxima de cada veículo: 60.

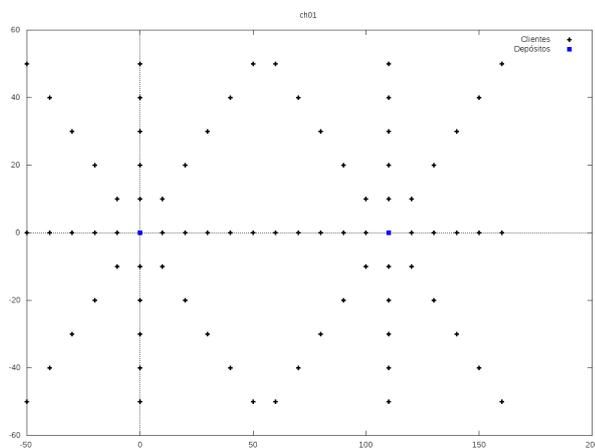


Figura A.12: Mapa - Instância ch01

A.3.2 Instância ch02

- 5 veículos;
- 80 clientes;
- 2 depósitos;
- Duração máxima do serviço: 200;
- Capacidade máxima de cada veículo: 60.

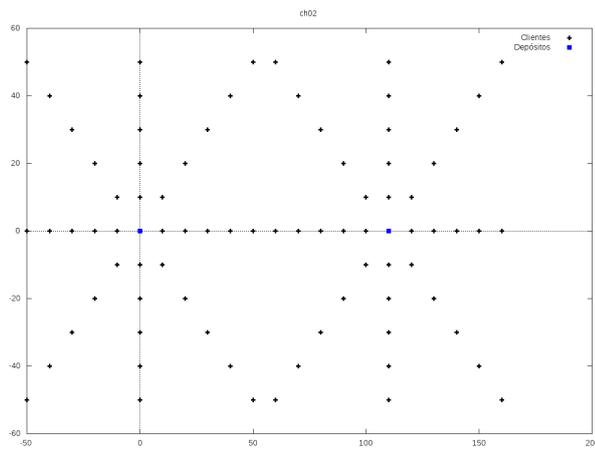


Figura A.13: Mapa - Instância ch02

A.3.3 Instância ch03

- 5 veículos;
- 80 clientes;
- 2 depósitos;
- Duração máxima do serviço: 180;
- Capacidade máxima de cada veículo: 60.

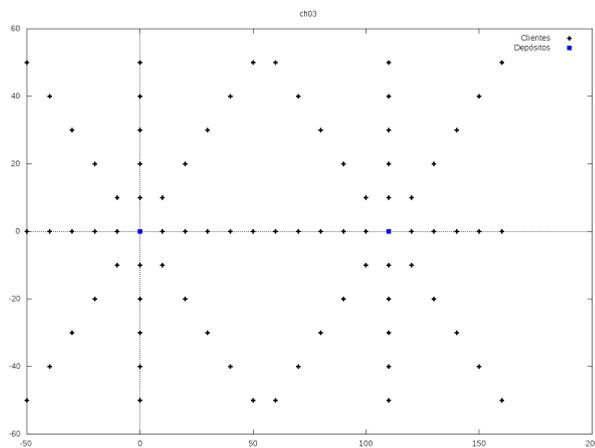


Figura A.14: Mapa - Instância ch03

A.3.4 Instância ch04

- 5 veículos;
- 160 clientes;
- 4 depósitos;
- Duração máxima do serviço: 0;
- Capacidade máxima de cada veículo: 60.

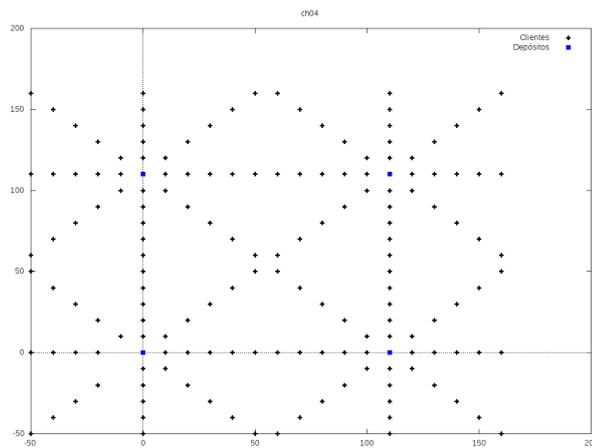


Figura A.15: Mapa - Instância ch04

A.3.5 Instância ch05

- 5 veículos;
- 160 clientes;
- 4 depósitos;
- Duração máxima do serviço: 200;
- Capacidade máxima de cada veículo: 60.

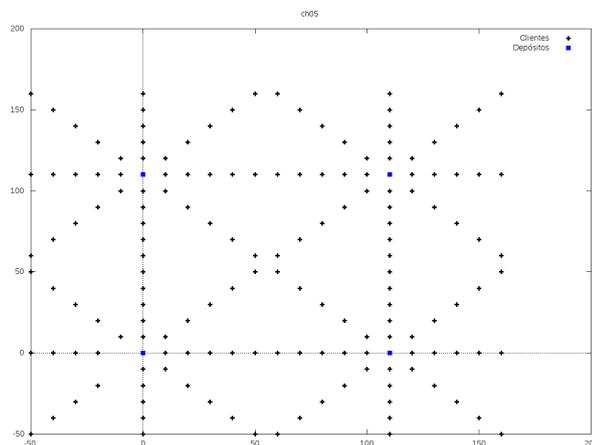


Figura A.16: Mapa - Instância ch05

A.3.6 Instância ch06

- 5 veículos;
- 160 clientes;
- 4 depósitos;
- Duração máxima do serviço: 180;
- Capacidade máxima de cada veículo: 60.

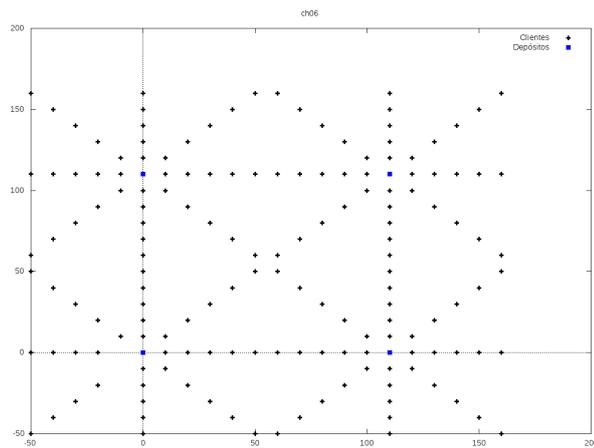


Figura A.17: Mapa - Instância ch06

A.3.7 Instância ch07

- 5 veículos;
- 240 clientes;
- 6 depósitos;
- Duração máxima do serviço: 0;
- Capacidade máxima de cada veículo: 60.

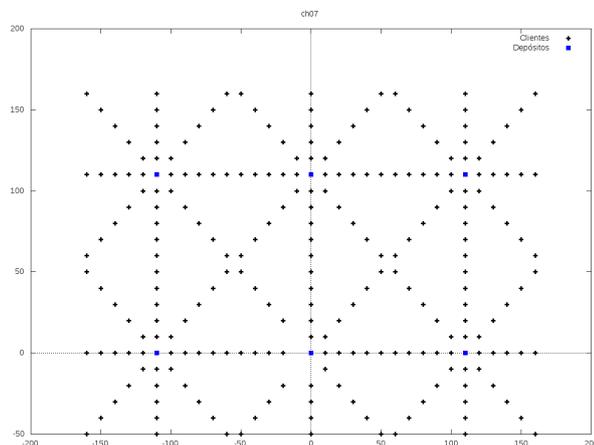


Figura A.18: Mapa - Instância ch07

A.3.8 Instância ch08

- 5 veículos;
- 240 clientes;
- 6 depósitos;
- Duração máxima do serviço: 200;
- Capacidade máxima de cada veículo: 60.

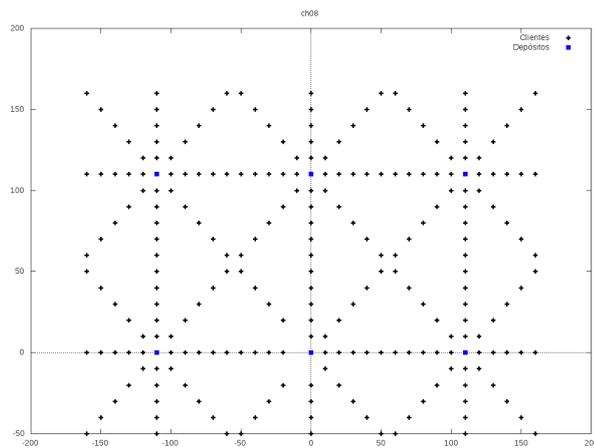


Figura A.19: Mapa - Instância ch08

A.3.9 Instância ch09

- 5 veículos;
- 240 clientes;
- 6 depósitos;
- Duração máxima do serviço: 180;
- Capacidade máxima de cada veículo: 60.

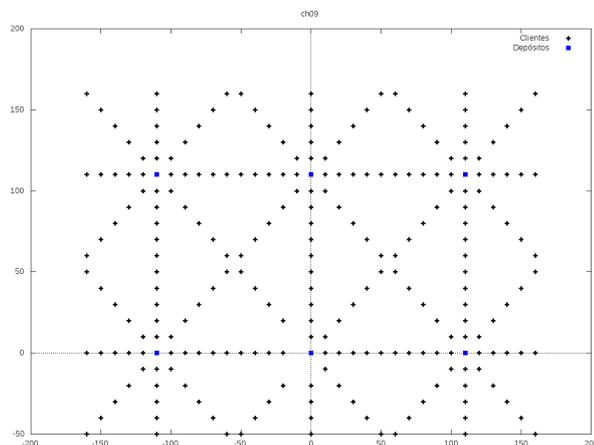


Figura A.20: Mapa - Instância ch09

A.3.10 Instância ch10

- 5 veículos;
- 360 clientes;
- 9 depósitos;
- Duração máxima do serviço: 0;
- Capacidade máxima de cada veículo: 60.

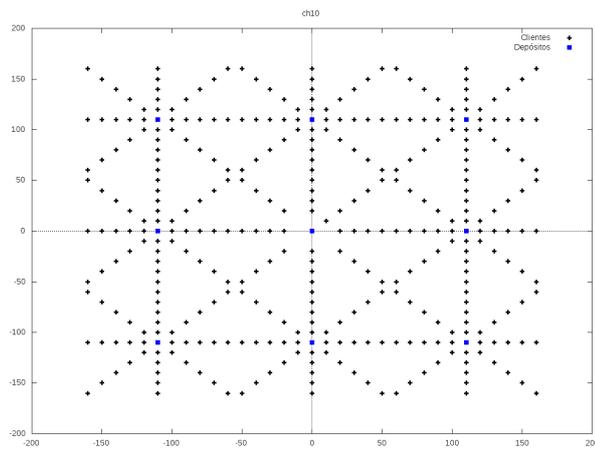


Figura A.21: Mapa - Instância ch10

A.3.11 Instância ch11

- 5 veículos;
- 360 clientes;
- 9 depósitos;
- Duração máxima do serviço: 200;
- Capacidade máxima de cada veículo: 60.

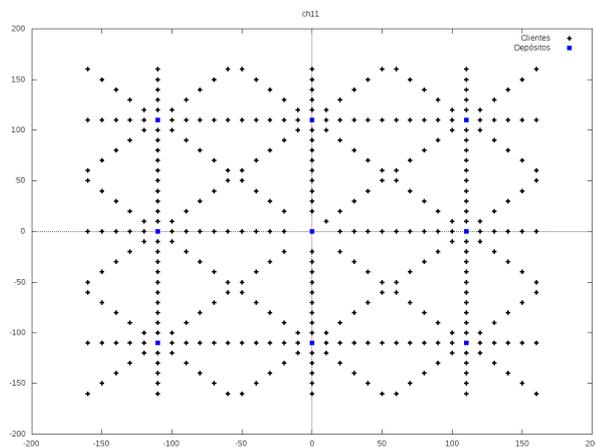


Figura A.22: Mapa - Instância ch11

A.3.12 Instância ch12

- 5 veículos;
- 360 clientes;
- 9 depósitos;
- Duração máxima do serviço: 180;
- Capacidade máxima de cada veículo: 60.

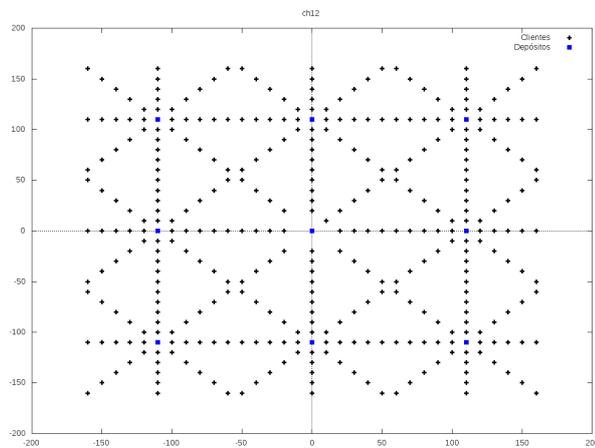


Figura A.23: Mapa - Instância ch12

A.4 Instâncias propostas por Cordeau et al. (1997):

A.4.1 Instância co01

- 1 veículo;
- 48 clientes;
- 4 depósitos;
- Duração máxima do serviço: 500;
- Capacidade máxima de cada veículo: 200.

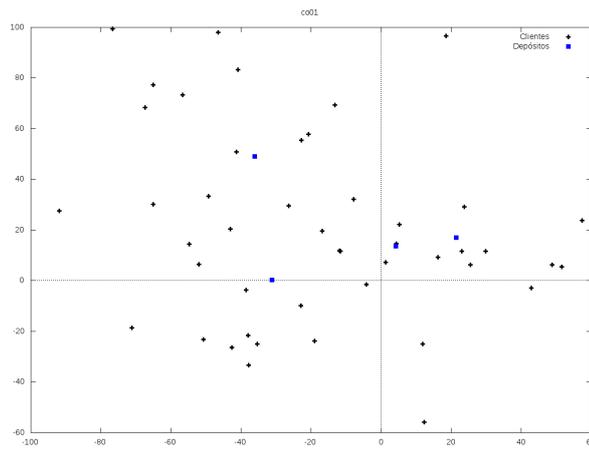


Figura A.24: Mapa - Instância co01

A.4.2 Instância co02

- 2 veículos;
- 96 clientes;
- 2 depósitos;
- Duração máxima do serviço: 480;
- Capacidade máxima de cada veículo: 195.

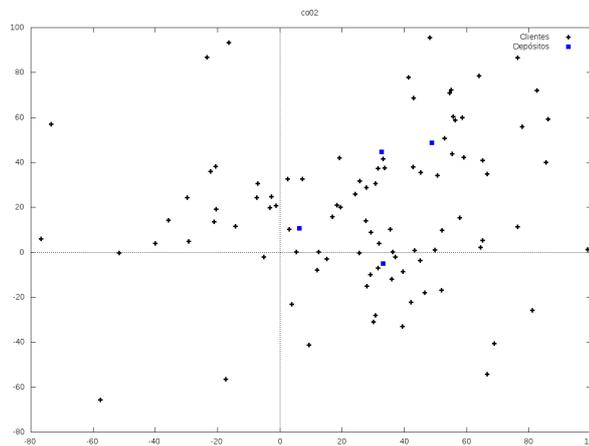


Figura A.25: Mapa - Instância co02

A.4.3 Instância co03

- 3 veículos;
- 144 clientes;
- 4 depósitos;
- Duração máxima do serviço: 460;

- Capacidade máxima de cada veículo: 190.

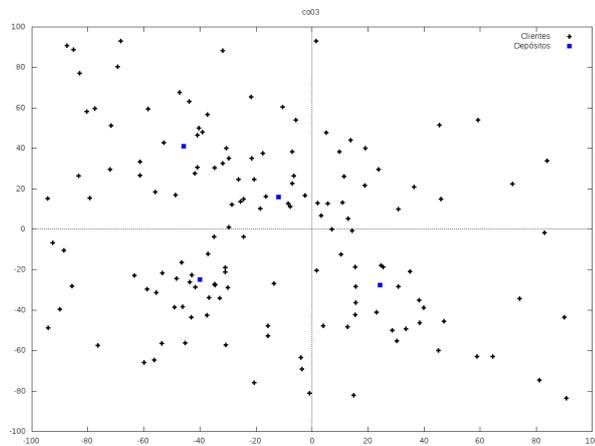


Figura A.26: Mapa - Instância co03

A.4.4 Instância co04

- 4 veículos;
- 192 clientes;
- 4 depósitos;
- Duração máxima do serviço: 440;
- Capacidade máxima de cada veículo: 185.

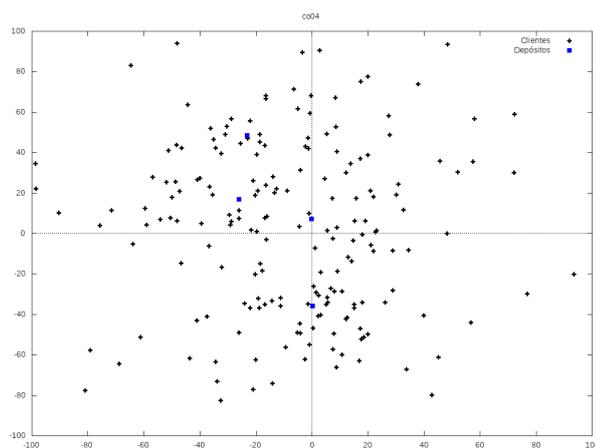


Figura A.27: Mapa - Instância co04

A.4.5 Instância co05

- 5 veículos;
- 240 clientes;

- 4 depósitos;
- Duração máxima do serviço: 420;
- Capacidade máxima de cada veículo: 180.

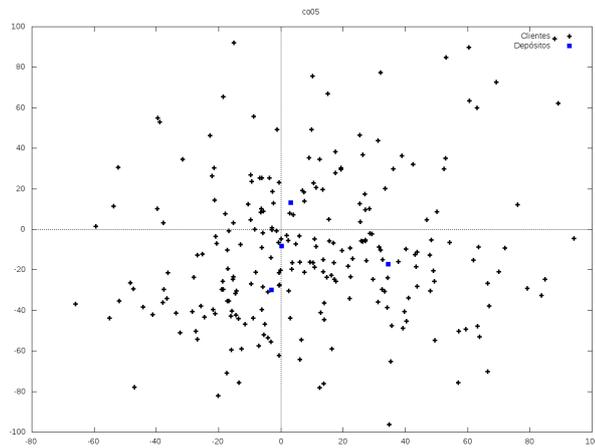


Figura A.28: Mapa - Instância co05

A.4.6 Instância co06

- 6 veículos;
- 288 clientes;
- 4 depósitos;
- Duração máxima do serviço: 400;
- Capacidade máxima de cada veículo: 175.

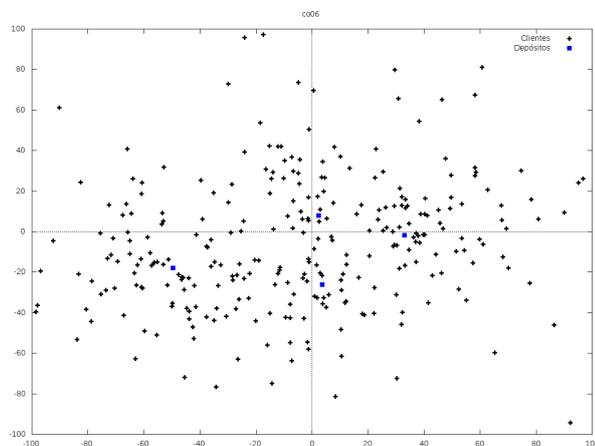


Figura A.29: Mapa - Instância co06

A.4.7 Instância co07

- 1 veículo;
- 72 clientes;
- 6 depósitos;
- Duração máxima do serviço: 500;
- Capacidade máxima de cada veículo: 200.

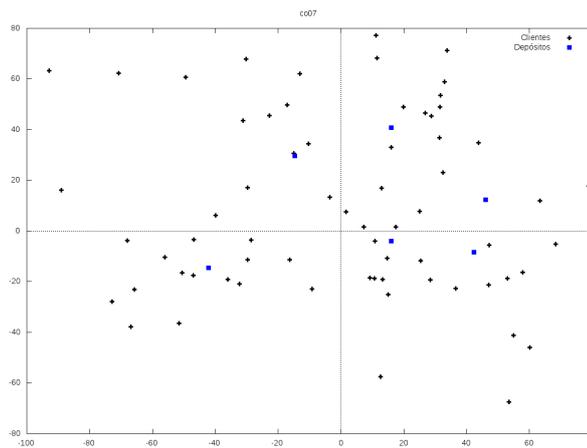


Figura A.30: Mapa - Instância co07

A.4.8 Instância co08

- 2 veículos;
- 144 clientes;
- 6 depósitos;
- Duração máxima do serviço: 475;
- Capacidade máxima de cada veículo: 190.

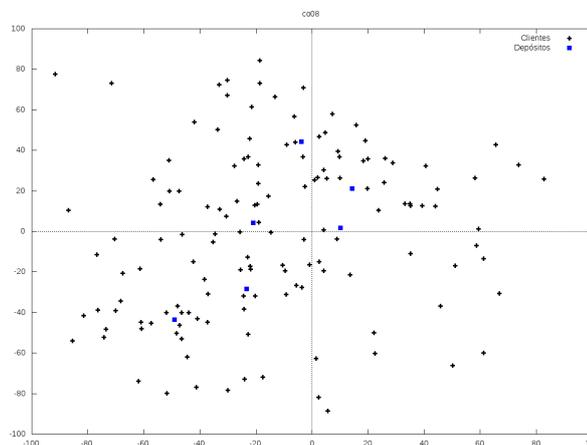


Figura A.31: Mapa - Instância co08

A.4.9 Instância co09

- 3 veículos;
- 216 clientes;
- 6 depósitos;
- Duração máxima do serviço: 450;
- Capacidade máxima de cada veículo: 180.

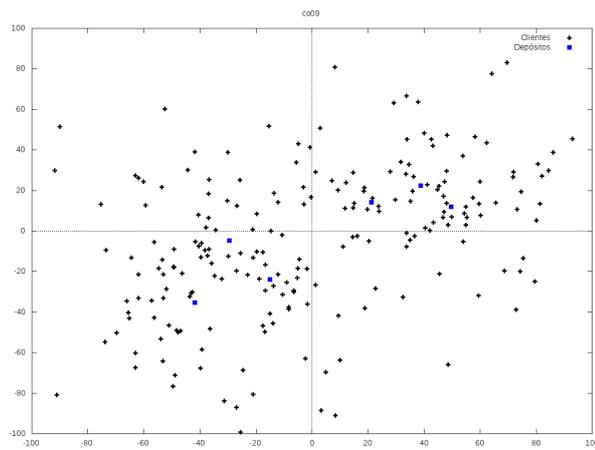


Figura A.32: Mapa - Instância co09

A.4.10 Instância co10

- 2 veículos;
- 288 clientes;
- 6 depósitos;
- Duração máxima do serviço: 425;
- Capacidade máxima de cada veículo: 170.

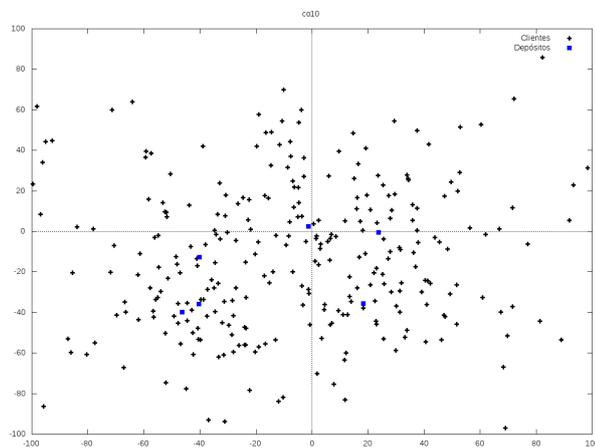


Figura A.33: Mapa - Instância co10