

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTO ACADÊMICO DE INFORMÁTICA
CURSO DE ENGENHARIA DE COMPUTAÇÃO

MARCIO ANTONIO COLTRO SCZEPANSKI

**OTIMIZAÇÃO DO PROBLEMA DE ROTEAMENTO DE VEÍCULOS
CAPACITADOS UTILIZANDO A META-HEURÍSTICA COLÔNIA DE
FORMIGAS**

TRABALHO DE CONCLUSÃO DE CURSO

PATO BRANCO

2018

MARCIO ANTONIO COLTRO SCZEPANSKI

**OTIMIZAÇÃO DO PROBLEMA DE ROTEAMENTO DE VEÍCULOS
CAPACITADOS UTILIZANDO A META-HEURÍSTICA COLÔNIA DE
FORMIGAS**

Trabalho de Conclusão de Curso de graduação, apresentado à disciplina de Trabalho de Conclusão de Curso 2, do Curso Superior de Engenharia de Computação do Departamento Acadêmico de Informática - DAINF - da Universidade Tecnológica Federal do Paraná - UTFPR, como requisito parcial para obtenção do título de “Engenheiro de Computação”.

Orientador: Prof. Dr. Marco Antônio de Castro
Barbosa

PATO BRANCO

2018



TERMO DE APROVAÇÃO

Às 8 horas e 20 minutos do dia 05 de junho de 2018, na sala V008, da Universidade Tecnológica Federal do Paraná, Câmpus Pato Branco, reuniu-se a banca examinadora composta pelos professores Marco Antonio de Castro Barbosa (orientador), Dalcimar Casanova e Jean Paulo Martins para avaliar o trabalho de conclusão de curso com o título **Otimização do Problema de Roteamento de Veículos Capacitado Utilizando a Metaheurística Colônia de Formigas**, do aluno **Marcio Antonio Coltro Sczepanski**, matrícula 01270737, do curso de Engenharia de Computação. Após a apresentação o candidato foi arguido pela banca examinadora. Em seguida foi realizada a deliberação pela banca examinadora que considerou o trabalho aprovado.

Marco Antonio de Castro Barbosa
Orientador (UTFPR)

Dalcimar Casanova
(UTFPR)

Jean Paulo Martins
(UTFPR)

Profa. Beatriz Terezinha Borsoi
Coordenador de TCC

Prof. Pablo Guterio Cavalcanti
Coordenador do Curso de
Engenharia de Computação

A Folha de Aprovação assinada encontra-se na Coordenação do Curso.

RESUMO

SCZEPANSKI, Marcio Antonio Coltro. OTIMIZAÇÃO DO PROBLEMA DE ROTEAMENTO DE VEÍCULOS CAPACITADOS UTILIZANDO A META-HEURÍSTICA COLÔNIA DE FORMIGAS. 46 f. Trabalho de Conclusão de Curso – Curso de Engenharia de Computação, Universidade Tecnológica Federal do Paraná. Pato Branco, 2018.

O Problema de Roteamento de Veículos (PRV) é um dos problemas de otimização mais estudados no ramo de Pesquisa Operacional devido a sua alta aplicabilidade. Porém, por fazer parte da categoria de problemas NP -difíceis, em muitos casos é necessário o uso de técnicas não exatas para geração de soluções. Neste trabalho é proposta a implementação de duas meta-heurísticas: a Colônia de Formigas e a Colônia de Formigas Elitista com finalidade de obter soluções aproximadas em tempo polinomial para o Problema de Roteamento de Veículos Capacitados.

Palavras-chave: Problema de Roteamento de Veículos, Meta-Heurística, Colônia de Formigas, Otimização Combinatória

ABSTRACT

SCZEPANSKI, Marcio Antonio Coltro. CAPACITATED VEHICLE ROUTING PROBLEM OPTIMIZATION USING THE ANT COLONY META-HEURISTIC . 46 f. Trabalho de Conclusão de Curso – Curso de Engenharia de Computação, Universidade Tecnológica Federal do Paraná. Pato Branco, 2018.

The Vehicle Routing Problem (VRP) is one of the most studied optimization problems in Operational Research due to its high practical applicability. However, the problem is contained in the category of difficult $NP - hard$, in many cases it is necessary to use non-exact techniques to generate solutions. This paper proposes the implementation of two meta-heuristics: the Ant Colony and the Elitist Ant Colony to obtain polynomial time solutions for the Capacitated Vehicle Routing Problem.

Keywords: Ant Colony, Combinatorial Optimization, Vehicle Routing Problem, Meta-Heuristics

LISTA DE FIGURAS

FIGURA 1	– Ilustração que demonstra como são as classe P e NP caso $P \subset NP$ e $P = NP$ respectivamente Fonte:(SIPSER, 2007).	11
FIGURA 2	– Universo NP com adição dos problemas $NP - completo$ para hipótese $P \subset NP$	12
FIGURA 3	– Formigas gerando um caminho de menor distância tendo que foi adicionado um obstáculo	15
FIGURA 4	– Uma instância do PRV e um exemplo de solução, respectivamente	18
FIGURA 6	– Diagrama ilustrativo para o funcionamento do método guloso	22
FIGURA 8	– Método Colônia de Formigas com variação dos parâmetros e $\rho = 0.2$	31
FIGURA 9	– Método Colônia de Formigas com variação dos parâmetros e $\rho = 0.5$	31
FIGURA 10	– Método Colônia de Formigas com variação dos parâmetros e $\rho = 0.8$	32
FIGURA 11	– Método Colônia de Formigas Elitista com variação dos parâmetros $\rho = 0.2$ e $n_{fe} = 2$	33
FIGURA 12	– Método Colônia de Formigas Elitista com variação dos parâmetros $\rho = 0.2$ e $n_{fe} = 5$	33
FIGURA 13	– Método Colônia de Formigas Elitista com variação dos parâmetros $\rho = 0.5$ e $n_{fe} = 2$	34
FIGURA 14	– Método Colônia de Formigas Elitista com variação dos parâmetros $\rho = 0.5$ e $n_{fe} = 5$	34
FIGURA 15	– Método Colônia de Formigas Elitista com variação dos parâmetros $\rho = 0.8$ e $n_{fe} = 2$	35
FIGURA 16	– Método Colônia de Formigas Elitista com variação dos parâmetros $\rho = 0.8$ e $n_{fe} = 5$	35
FIGURA 17	– Parâmetros aplicados na instância 2	37
FIGURA 18	– Parâmetros aplicados na instância 3	37
FIGURA 19	– Parâmetros aplicados na instância 4	38
FIGURA 20	– Parâmetros aplicados na instância 5	38
FIGURA 21	– Parâmetros aplicados na instância 11	39
FIGURA 22	– Parâmetros aplicados na instância 12	39

SUMÁRIO

1 INTRODUÇÃO	6
1.1 PROBLEMA	7
1.2 SOLUÇÃO PROPOSTA	7
1.3 OBJETIVOS	7
1.3.1 Objetivo Geral	8
1.3.2 Objetivos Específicos	8
1.4 JUSTIFICATIVA	8
1.5 ESTRUTURA DO TRABALHO	8
2 REFERENCIAL TEÓRICO	10
2.1 TIPOS DE PROBLEMAS	10
2.2 CLASSES DE PROBLEMAS P E NP	11
2.2.1 Classe NP-Completo e NP-Difícil	11
2.3 ESTRATÉGIAS DE SOLUÇÃO PARA PROBLEMAS INTRATÁVEIS	12
2.3.1 Métodos Exatos	13
2.3.2 Heurísticas e Meta-Heurísticas	13
2.3.2.1 Método Guloso	13
2.3.2.2 ALGORITMO GENÉTICO	13
2.3.2.3 Colônia de Formigas	14
2.3.2.4 COLÔNIA DE FORMIGAS ELITISTA	17
2.4 PROBLEMA DE ROTEAMENTO DE VEÍCULOS	18
3 MÉTODO PROPOSTO	20
3.1 MODELO MATEMÁTICO DO PROBLEMA DE ROTEAMENTO DE VEÍCULOS CAPACITADOS	20
3.2 SOLUÇÕES PROPOSTAS	21
3.2.1 Algoritmo Guloso	22
3.2.2 Colônia de Formigas	22
3.2.3 Colônia de Formigas Elitista	26
3.3 TESTES E APRIMORAMENTOS	28
3.4 DESCRIÇÃO DAS INSTÂNCIAS DE TESTE	28
4 RESULTADOS	30
4.1 AVALIAÇÃO DOS RESULTADOS ENCONTRADOS COM A VARIAÇÃO DOS PARÂMETROS	30
4.2 COMPARAÇÃO DOS MELHORES RESULTADOS OBTIDOS COM OS DISPONÍVEIS NA LITERATURA	40
4.3 DISCUSSÃO SOBRE OS RESULTADOS OBTIDOS	41
5 CONCLUSÕES	43
5.1 TRABALHOS FUTUROS	43
REFERÊNCIAS	44

1 INTRODUÇÃO

Algoritmos de otimização são métodos de busca cujo objetivo é encontrar a melhor solução para um problema (COOK et al., 1997). Sua definição é simples, mas inúmeros problemas do mundo real caracterizam-se como problemas de otimização. Por exemplo, roteamento de veículos de socorro e em logística de empresas para realização de entregas, posicionamento de radares e antenas de comunicação, escalonamento de trabalhadores dentre outros.

Grande parte desses problemas pertencem a classe NP-Completo. Esse termo foi introduzido no início dos anos 70, e simboliza o abismo da intratabilidade inerente que os projetistas de algoritmos encontram ao tentar solucionar problemas maiores e mais complexos (GAREY; JOHNSON, 1979). Estes problemas podem ser relacionados a clássicos problemas computacionais, tais como: Problema de Roteamento de Veículos (PRV) (no caso dos veículos de socorro e logística de empresas), Problema do Caixeiro Viajante (PCV) (para customização de rotas), Problema de Cobertura Mínima de Vértices (PCMV) (no caso de posicionamento de radares, antenas, câmeras de segurança), Problema da Mochila (PDM) (usado para investimento de capital), dentre milhares de outros problemas (GAREY; JOHNSON, 1979).

Existem muitas técnicas que podem ser utilizadas na solução do PRV, porém elas possuem limitações, como os métodos exatos, que no fim da sua execução, sempre fornecem uma solução ótima mas com tempo de execução exponencial, o que limita seu uso para instâncias pequenas. São exemplos de técnicas exatas: *backtracking*, *branch-and-bound* e *branch-and-cut* (CORMEN et al., 2002). Outra alternativa para buscar soluções para o PRV é o uso do paralelismo, porém sua utilidade está atrelada à construção de programas que executam em computadores massivamente paralelos (ASANOVIC et al., 2009). Outra alternativa são as meta-heurísticas, ao contrário dos métodos exatos que garantem uma solução ótima para um dado problema, meta-heurísticas tendem a fornecer uma solução boa, sem garantia de otimalidade mas, diminuindo consideravelmente a quantidade de tempo necessária para obtenção de uma resposta (TALBI, 2009a). Exemplos dessas técnicas são: Colônia de Formigas (DORIGO; STÜTZLE, 2004), GRASP (*Greedy Randomized Adaptive Search Procedure*)

(RESENDE; RIBEIRO, 2003), Têmpera Simulada (*Simulated Annealing*) (KIRKPATRICK C. D. GELATT, 1983).

1.1 PROBLEMA

Problemas de roteamento de veículos consistem, basicamente, no atendimento de um conjunto de consumidores por intermédio de uma frota de veículos que partem de um ou mais pontos (GOLDBARG; LUNA, 2000). O PRV pode ser construído em um ambiente com muitas restrições, por exemplo, cada veículo possui uma capacidade diferente e cada consumidor possui uma demanda específica, fazendo com que o somatório de todas as demandas em uma rota não possa extrapolar a capacidade dos veículos destinados a ela.

A variação do PRV tratada neste trabalho é o Problema de Roteamento de Veículos Capacitados, o qual apresenta como restrição que o somatório das cargas entregues por cada veículo, que não pode ultrapassar sua capacidade, também tem como restrição a homogeneidade da frota.

1.2 SOLUÇÃO PROPOSTA

Tendo que PRVs práticos requerem respostas de boa qualidade o mais rápido possível, como no roteamento de veículos de socorro que envolvem situações em que vidas podem estar em risco e rotas viáveis podem deixar de existir, como congestionamentos ou acidentes na estradas, métodos heurísticos e meta-heurísticos são as estratégias mais indicadas para resolver estes problemas.

Em função do apresentado, é proposta a implementação de um sistema computacional que apresente uma solução de boa qualidade, tanto em termos de tempo quanto em distância percorrida pelos veículos.

1.3 OBJETIVOS

Considerando a necessidade de respostas de boa qualidade em um tempo hábil para o PRVC, no texto que segue são apresentados os objetivos deste trabalho.

1.3.1 OBJETIVO GERAL

Implementar a meta-heurística Colônia de Formigas e uma extensão da mesma, o método Colônia de Formigas Elitista para obter uma solução polinomial e de melhor qualidade possível para o Problema de Roteamento de Veículos Capacitado com Frota Homogênea, para assim, realizar a comparação dos melhores resultados encontrados com variações de parâmetros em ambas as técnicas.

1.3.2 OBJETIVOS ESPECÍFICOS

- Implementar um método guloso para gerar soluções para o PRVC.
- Implementar a meta-heurística Colônica de Formigas para gerar soluções para o PRVC.
- Implementar a extensão Colônia de Formigas Elitista para gerar soluções para o PRVC.
- Realizar experimentação com os parâmetros disponíveis em cada implementação buscando a configuração que gere os melhores resultados.
- Comparar dos resultados encontrados com resultados da literatura.

1.4 JUSTIFICATIVA

A principal justificativa para realização deste trabalho é a alta aplicabilidade prática do PRV, na maior parte das indústrias, a atividade de transporte representa um dos elementos mais importantes na composição do custo logístico. Nas nações desenvolvidas, os fretes costumam absorver até 60 % do gasto logístico total (NOVAES, 2007).

A escolha das técnicas de solução implementadas se baseiam na intuitiva possibilidade de paralelização das mesmas e na característica adaptativa dos métodos, ou seja, fazendo com que sejam aplicáveis a problemas reais, rotas as quais podem não ser estáticas durante a procura de uma solução.

1.5 ESTRUTURA DO TRABALHO

O segundo capítulo apresenta a revisão bibliográfica, sendo esta desdobrada em uma visão geral acerca do universo de complexidades de problemas, estratégias para solução de problemas intratáveis e sobre o Problema de Roteamento de Veículos.

O Capítulo Três contempla o desenvolvimento do trabalho. Na descrição do problema será definido o modelo matemático com todas as restrições para o Problema de Roteamento de Veículos Capacitados. Em seguida, será descrito o desenvolvimento do procedimento e relatados os aprimoramentos efetuados. Na descrição das unidades de análises serão explicitadas as instâncias de teste, retiradas da literatura.

O Capítulo Quatro apresenta, os resultados obtidos com variações dos parâmetros encontrados em cada método. Após esta apresentação são realizadas comparações entre as soluções obtidas, é então escolhida a configuração de parâmetros com melhores resultados, os quais são comparados com os encontrados na literatura.

As conclusões do trabalho e sugestões para trabalhos futuros são apresentadas no Capítulo Cinco

2 REFERENCIAL TEÓRICO

Neste capítulo serão apresentados conceitos básicos para completo entendimento dos métodos implementados e resultados obtidos.

2.1 TIPOS DE PROBLEMAS

Define-se problema como uma questão qualquer que necessita de uma resposta, geralmente o mesmo possui vários parâmetros ou variáveis, cujos valores não são especificados. Um problema é especificado por: (1) uma descrição de todos os parâmetros do mesmo, e (2) uma declaração de quais propriedades da resposta, ou solução, são necessárias para o satisfazer (GAREY; JOHNSON, 1979).

Um problema de decisão caracteriza-se como aquele que dada uma entrada qualquer a resposta apresenta apenas duas possibilidades, sendo elas “sim” ou “não”. Pode-se dizer também que a solução pode ser representada por apenas um bit 0 ou 1 (GOODRICH; TAMASSIA, 2014). Por exemplo, um problema que seja necessário descobrir se existe ou não um Circuito Hamiltoniano é um problema de decisão (TOSCANI; VELOSO, 2009). Na matemática aplicada e na Teoria da Computação, otimização combinatória é um tópico que consiste em encontrar um objeto ótimo dentro de um conjunto finito de objetos (COOK et al., 1997).

Por exemplo, o Problema de Roteamento de Veículos é um problema de otimização (TOSCANI; VELOSO, 2009). A função de complexidade de tempo para um algoritmo expressa seus requisitos temporais fornecendo para cada possível comprimento de entrada, a maior quantidade de tempo necessária do algoritmo para solucionar um problema para uma determinada instância (GAREY; JOHNSON, 1979).

2.2 CLASSES DE PROBLEMAS P E NP

“A classe de complexidade P é o conjunto de todos os problemas de decisão (ou linguagens) L que podem ser decididos em tempo polinomial no pior caso.” (GOODRICH; TAMASSIA, 2014). Mais especificamente, são problemas que podem ser resolvidos em tempo $O(n^k)$ para alguma constante k , onde n é o tamanho da entrada para o problema (CORMEN et al., 2009).

A classe NP consiste nos problemas que são “verificáveis” em tempo polinomial. O que significa que, existindo algum tipo de “certificado” de uma solução, poderia-se verificar se o certificado é correto em tempo polinomial para o tamanho da entrada do problema (CORMEN et al., 2009). Existe uma forma determinística de se definir um problema NP, a qual diz que se um problema pode ser verificado através de uma máquina de Turing determinística ele pertence a classe NP, ou seja, dada uma entrada é possível verificar se a ela corresponde a uma solução do problema em tempo polinomial (GAREY; JOHNSON, 1979).

Um dos maiores problemas não resolvidos da ciência da computação e matemática contemporânea, é se as classes P e NP são iguais. Qualquer problema em P também está em NP visto que, se um problema está em P , é possível resolvê-lo em tempo polinomial sem nem mesmo ter um “certificado”, a questão em aberto é se P é ou não um subconjunto próprio de NP (CORMEN et al., 2009). Uma forma de ilustrar essas duas classes nas duas configurações possíveis ($P \subset NP$ ou $P = NP$) é como na Figura 1.

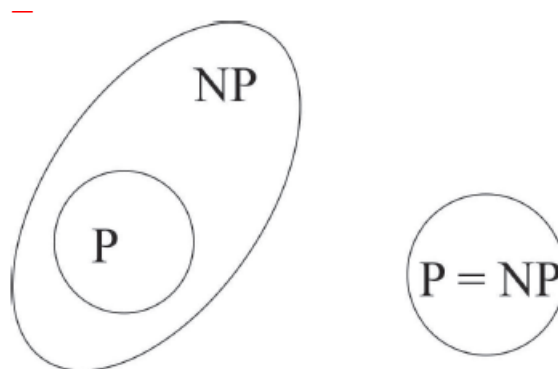


Figura 1 – Ilustração que demonstra como são as classe P e NP caso $P \subset NP$ e $P = NP$ respectivamente
 Fonte:(SIPSER, 2007).

2.2.1 CLASSE NP-COMPLETO E NP-DIFÍCIL

Essa classe de problemas é considerada como “os problemas mais difíceis em NP ”, pois se existir uma solução em tempo polinomial para um único problema em NP-Completo,

todos os problemas em NP poderão ser resolvidos, mas se um único problema em NP for intratável, então todos os problemas em $NP - Completo$ também o serão (GAREY; JOHNSON, 1979). Formalmente uma linguagem L é definida como $NP - Completa$, se ela pertence a NP , e para todas as outras linguagens L' que pertencerem a NP , L' será equivalente a L . Informalmente, um problema de decisão Π é $NP - Completo$ se Π pertence a NP , e para todos os outros problemas de decisão Π' pertencentes a NP , Π' equivale a Π (GAREY; JOHNSON, 1979).

Através das definições vistas anteriormente, e assumindo que P é diferente de NP , pode-se ter uma visão nova do universo NP , como ilustra a Figura 2.

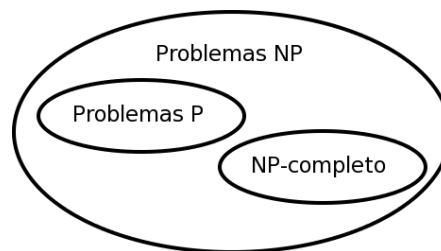


Figura 2 – Universo NP com adição dos problemas $NP - completo$ para hipótese $P \subset NP$

Então, formalmente, para provar que um problema Π é $NP - Completo$ é necessário mostrar que:

- Π pertence a NP .
- Algum problema Π' NP-Completo seja transformável, em tempo polinomial, a Π (GAREY; JOHNSON, 1979).

Outra classe importante a ser tratada é a classe de problemas NP -Difícil, esta classe caracteriza-se por problemas que não podem ser resolvidos de forma determinística em tempo polinomial, a não ser que seja provado que $P=NP$. Para saber se um problema Π é NP -difícil, basta reduzir qualquer problema NP Π' a Π . Se $\Pi' \propto \Pi$ então Π é um NP -difícil (TOSCANI; VELOSO, 2009).

2.3 ESTRATÉGIAS DE SOLUÇÃO PARA PROBLEMAS INTRATÁVEIS

De acordo com a complexidade do problema, o problema pode ser resolvido por um método exato ou um método aproximado. Para problemas da classe NP -Completo, algoritmos exatos são inviáveis em alguns casos devido ao tempo computacional tomado, tendo-se então a necessidade de utilizar os métodos aproximados. Métodos aproximados geram soluções em um

tempo razoável para uso prático, mas não há garantia de encontrar uma solução ótima global (CORMEN et al., 2009)(TALBI, 2009b).

2.3.1 MÉTODOS EXATOS

Métodos exatos garantem encontrar uma solução ótima para cada instância finita de problemas de otimização combinatória. Mas para problemas NP-Difíceis não existe solução em tempo polinomial (BLUM; ROLI, 2003). São exemplos de técnicas exatas: *backtracking*, *branch-and-bound* e *branch-and-cut* (CORMEN et al., 2009).

2.3.2 HEURÍSTICAS E META-HEURÍSTICAS

Heurísticas são critérios, métodos ou princípios para decidir quais dentre vários cursos de ação prometem ser os mais eficazes para atingir algum objetivo (PEARL, 1984).

2.3.2.1 MÉTODO GULOSO

Esse método consiste em iniciar com um conjunto de solução vazio e constrói-se então uma solução atribuindo valores para uma variável de decisão por vez, até uma solução completa seja gerada. Entretanto essa abordagem nem sempre tende para uma solução ótima (GOODRICH; TAMASSIA, 2014).

Tendo que métodos exatos são muito caros para problemas do mundo real e heurísticas tradicionais tendem a cair em ótimos locais é necessária a projeção de algoritmos capazes de escapar do ótimo local (MICHALEWICZ; FOGEL, 2013). Meta-heurísticas, em sua definição original, são métodos de solução que orquestram a interação entre procedimentos de melhoria local e estratégias de alto nível para criar um processo capaz de escapar de ótimos locais e realizando uma pesquisa robusta no espaço da solução (GLOVER; KOCHENBERGER, 2003). São exemplos de Meta-heurísticas : *GRASP*, Algoritmo Genético (*Genetic Algorithm*), Têmpera Simulada (*simulated annealing*), Colônia de Formigas (*Ant Colony Optimization*) (GLOVER; KOCHENBERGER, 2003).

2.3.2.2 ALGORITMO GENÉTICO

Utilizado pela primeira vez por John Holland em 1975, o Algoritmo Genético é uma técnica que se baseia na *Teoria de Seleção Natural* de Darwin (GLOVER; KOCHENBERGER, 2006). A partir de uma população inicial, o algoritmo realiza cruzamentos, mutações e recombinações a fim de gerar melhores soluções.

Esse algoritmo mantém uma população inicial de indivíduos, $P(t) = x_t^1, \dots, x_t^n$ para a iteração t . Cada indivíduo representa uma solução em potencial para o problema. Cada solução x_t^i é avaliada e é medida como um “*fitness*”. Uma nova população na iteração $t + 1$ é então formada a partir da seleção dos indivíduos de acordo com seus *fitness*. Alguns membros da nova população sofrem transformações para que sejam criados novos membros para o grupo. Essa etapa de transformação são classificadas como unitárias, quando o indivíduo sofre uma mutação gerando um novo indivíduo, e de maior ordem, onde dois ou mais indivíduos são combinados formando um novo indivíduo como um processo de cruzamento (GLOVER; KOCHENBERGER, 2006). O pseudo código genérico para o Algoritmo Genético é apresentado a seguir:

Algorithm 1: Algoritmo Genético

```

1  $t = 0$ ;
2 Inicializa( $P(t)$ );
3 Avalia( $P(t)$ );
4 while Critério de parada não satisfeito do
5   |  $t = t + 1$ ; Selecione  $P(t)$  de  $P(t - 1)$ ; Mutação  $P(t)$ ; Cruzamento  $P(t)$ ; Avalia
   |  $P(t)$ ;
6 end while

```

2.3.2.3 COLÔNIA DE FORMIGAS

O método de otimização Colônia de Formigas (ACO) é uma aproximação meta-heurística para solução de problemas combinatórios de otimização difíceis (DORIGO; CARO, 1999). Algoritmos de formiga são sistemas multi-agentes no qual o comportamento de cada agente, chamado de *formiga artificial* ou *formiga*, é baseado no comportamento de formigas reais. Esse método é um dos que obtém melhores resultados na família de métodos de inteligência por enxame (BONABEAU et al., 1999) e foi primeiramente proposto por Marco Dorigo em 1992 na sua tese de Doutorado Dorigo e Caro (1999).

O exemplo biológico no qual o método é baseado foi obtido através do experimento feito por Goss utilizando uma colônia de formigas real. Estas formigas, conforme realizavam o caminho da colônia para o alimento e vice versa, depositam uma substância química chamada de *feromônio* no chão. Quando chegam a um ponto aonde é necessário realizar uma escolha de caminho, elas a fazem de uma forma probabilística baseada na quantidade de feromônio depositada em cada caminho. Este comportamento faz com que, o caminho com mais feromônio seja escolhido mais vezes, fazendo com que mais formigas depositem feromônio

neste caminho e consequentemente deixando ele mais atrativo para outras formigas (DORIGO; CARO, 1999).

O comportamento descrito anteriormente é mostrado na Figura 3, em (A), as formigas estão seguindo o caminho de menor distância, em (B) é adicionado um obstáculo, em (C) o comportamento inicial das formigas é mostrado sendo que a probabilidade de escolher o caminho para desviar o obstáculo inicialmente é a mesma, já que ainda não foi adicionado feromônio por nenhum dos caminhos, em (D) é demonstrado o que as formigas convergiram para o caminho de menor distância, pois como a trilha da direita é menor e consequentemente requer menos tempo de viagem, as formigas deixaram mais feromônio nesse caminho, e quanto mais formigas passam por esse caminho, maior será o nível de feromônio (TALBI, 2009a).

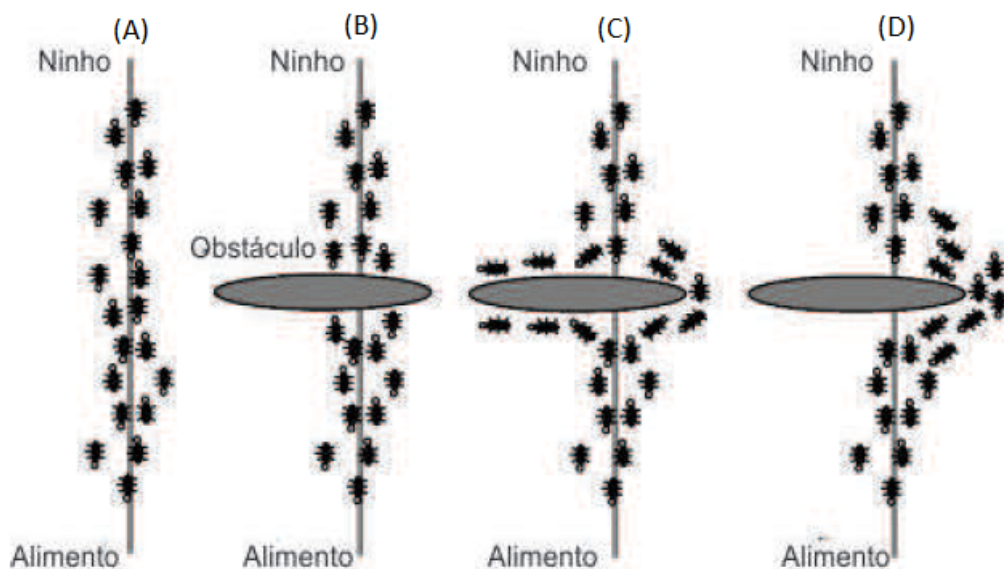


Figura 3 – Formigas gerando um caminho de menor distância tendo que foi adicionado um obstáculo

Informalmente, o comportamento das formigas em um algoritmo ACO pode ser resumido como: uma colônia de formigas se move de forma dessincronizada pelos estados adjacentes do problema, construindo soluções. Durante o percurso, as formigas deixam uma trilha de feromônio, o qual tem função análoga ao exemplo biológico. Conforme as formigas se movem, criam um conjunto de soluções. Quando uma formiga constrói uma solução, ela avalia a mesma e deposita um traço de feromônio baseado nessa avaliação nas arestas percorridas. Essa informação de feromônio afeta diretamente a pesquisa por soluções das próximas formigas (GLOVER; KOCHENBERGER, 2006). Além disso, a ação do feromônio diminui com o passar do tempo (através da evaporação).

O comportamento descrito pode ser convertido em um sistema computacional, como

proposto por Marco Dorigo e Lucas Maria Gambardella, o qual com algumas modificações pode ter como objetivo encontrar soluções para o PRV, primeiramente, começando de uma cidade inicial, que, no caso de Problemas de Roteamento de Veículos Capacitados é um dos depósitos de onde os veículos iniciam sua trajetória, uma formiga se move de cidade a cidade. Quando em uma cidade i , uma formiga k escolhe ir para uma cidade ainda não visitada j com a probabilidade dada por:

$$p_{ij}^k(t) = \frac{[\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}]^\beta}{\sum_{l \in N_i^k} [\tau_{il}(t)]^\alpha \cdot [\eta_{il}]^\beta} \text{ se } j \in N_i^k \quad (1)$$

η_{ij} é uma informação heurística disponível *a priori*, $\tau_{ij}(t)$ é a informação de feromônio, α e β são dois parâmetros que determinam a influência dos traços de feromônio e das informações heurísticas respectivamente, N_i^k é a vizinhança disponível para uma formiga k , a qual são todas as cidades ainda não visitadas por k . Os parâmetros α e β tem as seguintes influências no comportamento do algoritmo. Se $\alpha = 0$ a seleção de cidades é proporcional a $[\eta_{ij}]^\beta$ e as cidades mais próximas (com arestas de peso menor) tem mais chance de serem selecionadas, nesse caso o ACO corresponde a um algoritmo estocástico guloso clássico. Se $\beta = 0$ somente a ação do feromônio é considerada, o que de acordo com Dorigo e Stützle (2004), leva a um rápido surgimento de uma situação de estagnação na geração de novas soluções (GLOVER; KOCHENBERGER, 2006).

Após o término da construção das soluções por todas as formigas se dá início ao processo de depósito e evaporação da matriz de feromônios, e estes processos podem ser descritos das seguintes maneiras:

- Depósito:

$$\tau_{ij} = \tau_{ij} + Q/L_k \quad (2)$$

τ_{ij} são todas as arestas que compõem a solução encontrada pela formiga k , Q é uma constante e L_k é a distância total percorrida pela formiga em sua solução.

- Evaporação :

$$\tau_{ij} = (1 - \rho)\tau_{ij} \quad (3)$$

Processo o qual será feito apenas uma vez para todas as arestas τ_{ij} no final de cada iteração.

O componente estocástico de ACO permite que as formigas criem uma grande variedade de diferentes soluções e, portanto, pode explorar um número muito maior de soluções

do que heurísticas gulosas. Ainda mais importante, a experiência de busca das formigas pode ser usada para influenciar a construção de soluções em futuras iterações do algoritmo (GLOVER; KOCHENBERGER, 2003). O pseudocódigo dessa meta-heurística é apresentado a seguir:

Algorithm 2: Pseudo código Colônia de Formigas

```

1 Inicialização das matrizes de distância e feromônio;
2 while Critério de finalização não é satisfeito do
3   for Cada formiga do
4     |   Construção de soluções;
5   end for
6   Atualização do feromônio para arestas visitadas;
7   Evaporação;
8 end while

```

2.3.2.4 COLÔNIA DE FORMIGAS ELITISTA

Pesquisas sobre ACO mostram que é possível melhorar os resultados explorando as melhores soluções encontradas a cada iteração, porém, usar um sistema puramente elitista, o qual é um ACO aonde somente a formiga com melhor resultado, a que encontrou o melhor resultado dentre todas as iterações, ou local, a que pode alterar o feromônio, potencialmente agrava o problema de estagnação prematura da busca, ou seja, a busca tente a entrar em ótimos locais devido ao elitismo no depósito de feromônio. Então a chave para um melhor desempenho do ACO é combinar uma melhor exploração das melhores soluções encontradas durante as interações com um mecanismo eficiente de evitar a estagnação da busca (STÜTZLE; HOOS, 2000), um destes mecanismos pode ser a utilização de múltiplas formigas elitistas, o pseudo

código é apresentado a seguir:

Algorithm 3: Pseudo código Colônia de Formigas

```

1  inicialização das matrizes de distância e feromônio;
2  while Critério de finalização não é satisfeito do
3      for Cada formiga do
4          |   Construção de soluções;
5      end for
6      for  $n_{fe}$  formigas com melhor solução do
7          |   Atualização do feromônio para arestas visitadas;
8      end for
9      Evaporação;
10 end while

```

Aonde a diferença do método Colônia de Formigas reside na seleção das θ formigas que geraram as melhores soluções para serem responsáveis pelo depósito de feromônio.

2.4 PROBLEMA DE ROTEAMENTO DE VEÍCULOS

Um sistema de roteamento é definido como “um conjunto organizado de meios que objetiva o atendimento de demandas localizadas nos arcos e/ou nos vértices de alguma rede de transportes”(GOLDBARG; LUNA, 2000), O Problema de Roteamento de Veículos(PRV) é um dos problemas de otimização combinatória mais estudados devido a sua grande área de aplicação o objetivo do PRV é encontrar um *design* ótimo de rotas percorridas por uma frota de veículos para servir um conjunto de consumidores (BODIN et al., 1983), a Figura 4 demonstra uma instância e um exemplo de solução .

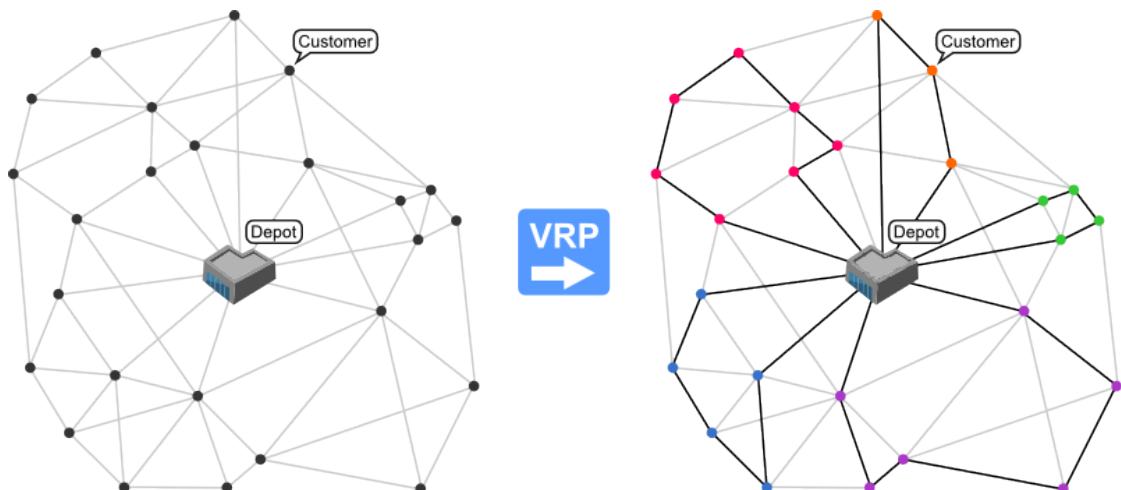


Figura 4 – Uma instância do PRV e um exemplo de solução, respectivamente

Desde que foi primeiramente apresentado por Danzig e Ramster, centenas de publicações foram feitas tendo como objetivo encontrar uma solução ótima ou aproximada para este problema. Devido a sua alta aplicabilidade, inúmeras variações deste problema surgiram com objetivo de se adaptar melhor a problemas reais, como a abordagem do PRV com janela de tempo (PRVJT) feita por Kolen, Kan e Trienekens (KOLEN et al., 1987) na qual uma frota fixa de veículos, com capacidade limitada e disponível em um único depósito precisava atender um certo grupo de clientes com uma certa demanda e cada cliente devia ser visitado dentro de um período específico de tempo. Outra variação é o Problema de Roteamento de Veículos com Entrega Dividida (PRVCED) (DROR et al., 1994), que é um PRV no qual cada consumidor pode ser visitado mais de um vez.

Como observado por Taillard (1993), PRVs são, na maioria das vezes, problemas NP-Difíceis e, portanto, técnicas que obtêm soluções exatas tendem a não ser eficientes, então abordagens de otimização combinatória e métodos heurísticos são mais recomendados. O Problema de Roteamento de Veículos Capacitado (PRVC) é um PRV no qual há uma quantidade fixa de veículos com capacidade uniforme que deve atender às demandas conhecidas de clientes. Ou seja, o PRVC é um PRV com uma restrição adicional de que todos os veículos tem uma capacidade uniforme com uma única mercadoria. O objetivo do problema é minimizar a quantidade necessária de veículos e o tempo necessário para completar o trajeto.

Utilizando técnicas exatas, Ladányi et al. (2001) apresenta um algoritmo *Branch, Cut and Prime* para encontrar soluções para problemas de otimização de larga escala de forma sequencial e paralela. Um exemplo de método heurístico para gerar soluções para o PRV é o algoritmo *Clark and Wright* apresentado em Clarke e Wright (1964), o qual consiste de uma heurística que pode ser aplicada ao PRV quando o número de veículos não é fixo. Quando duas rotas $(0, \dots, i, 0)$ e $(0, j, \dots, 0)$ podem ser fundidas em uma única rota, uma economia de distância $s_{ij} = c_{i0} + c_{0j} - c_{ij}$ é gerada. Em Tan et al. (2012) é apresentado um algoritmo Colônia de Formigas que utiliza heurísticas e métodos de busca local como estratégias de melhoramento das soluções geradas.

3 MÉTODO PROPOSTO

O Problema de Roteamento de Veículos Capacitados (PRVC) tratado neste trabalho pode ser definido da seguinte forma: Seja $G = (V, A)$ um grafo em que $V = 0, \dots, n$, é um conjunto de vértices que representa as cidades, ou clientes, a serem visitados pelos veículos com o vértice 0 sendo o depósito de onde os veículos começarão sua trajetória e n a quantidade de clientes a serem visitados, A é o conjunto de arestas do grafo no qual cada aresta (i, j) , $i \neq j$, é associada a uma matriz de distâncias $C = (c_{ij})$. Para este problema, assume-se que existem X veículos disponíveis do depósito, os quais possuem uma capacidade P . Para que se possa obter rotas que compõe as soluções do problema é necessário seguir as seguintes restrições.

- Cada vértice em V é visitado exatamente uma vez;
- Todas as rotas iniciam e terminam no depósito;
- A soma das demandas de qualquer rota feita pelo veículo não pode exceder a capacidade Q do veículo.

3.1 MODELO MATEMÁTICO DO PROBLEMA DE ROTEAMENTO DE VEÍCULOS CAPACITADOS

O PRV Capacitado abordado neste trabalho pode ser formulado seguindo o modelo matemático de índice triplo de fluxo de veículos adotado por Toth e Vigo (2001), apresentado a seguir:

$$\text{Minimizar } \sum_{i=0}^n \sum_{j=0}^n c_{ij} \sum_{k=0}^x x_{ijk} \quad (4)$$

$$\text{Sujeito a } \sum_{k=1}^K y_{ik} = 1 (i = 1, \dots, n) \quad (5)$$

$$\sum_{k=1}^K y_{0k} = K \quad (6)$$

$$\sum_{j=0}^n x_{ijk} = \sum_{j=0}^n x_{jik} = y_{ik} (i = 0, \dots, n), (k = 1, \dots, K) \quad (7)$$

$$\sum_{i=1}^n d_i y_{ik} \leq P (k = 1, \dots, K) \quad (8)$$

$$\sum_{i \in S} \sum_{j \in S} x_{ijk} \leq |S| - 1 (\forall S \subseteq V, 0, |S| \geq 2, k = 1, \dots, K) \quad (9)$$

$$x_{ijk} \in 0, 1 (i, j = 0, \dots, n; i \neq j), (k = 1, \dots, K) \quad (10)$$

$$y_{ij} \in 0, 1 (i, j = 0, \dots, n; i \neq j), (k = 1, \dots, K) \quad (11)$$

A função objetivo (4) busca a minimização da distância total percorrida, variável binária x_{ijk} , a qual contém as informações sobre às arestas disponíveis no grafo G , recebe 1 se o arco (i, j) é coberto pelo veículo k na sua solução, o que define que este arco não pode ser visitado novamente, 0 caso contrário. A variável binária y_{ik} , a qual contém as informações sobre os vértices disponíveis no grafo G , recebe valor 1 caso o vértice i seja visitado pelo veículo k em sua solução, implicando que este vértice não pode ser visitado novamente por nenhum veículo, 0 caso contrário. Portanto, a restrição (5) desta formulação impõe que cada vértice i seja visitado por apenas um veículo, a restrição (6) determina que K veículos devem deixar o depósito e a restrição (7) estabelece que o mesmo veículo k deve entrar e sair de um determinado vértice i . A carga especificada de cada veículo não pode ultrapassar a capacidade P , de acordo com a restrição (8). A restrição (9) elimina sub-circuitos ilegais, determinando que para qualquer subconjunto S de $V \setminus 0$ o número total de arcos $x_{ijk} (j, i \in S)$ pertencentes a solução ótima deve ser menor ou igual ao número de elementos de S menos um, as últimas duas restrições (10) e (11) indicam que as variáveis x_{ij} e y_{ik} são binárias.

3.2 SOLUÇÕES PROPOSTAS

Foram implementados três algoritmos que tem como objetivo gerar soluções para o PRVC, o algoritmo Colônia de Formigas, uma de suas variações, o algoritmo Colônia de Formigas Elitista e o algoritmo Guloso. Os algoritmos foram implementados utilizando a linguagem de programação C.

3.2.1 ALGORITMO GULOSO

O método guloso foi implementado com base na teoria abordada no Capítulo 2 Secção 2.3.2.1, e seu funcionamento pode ser ilustrado na Figura 6, apresentada a seguir.

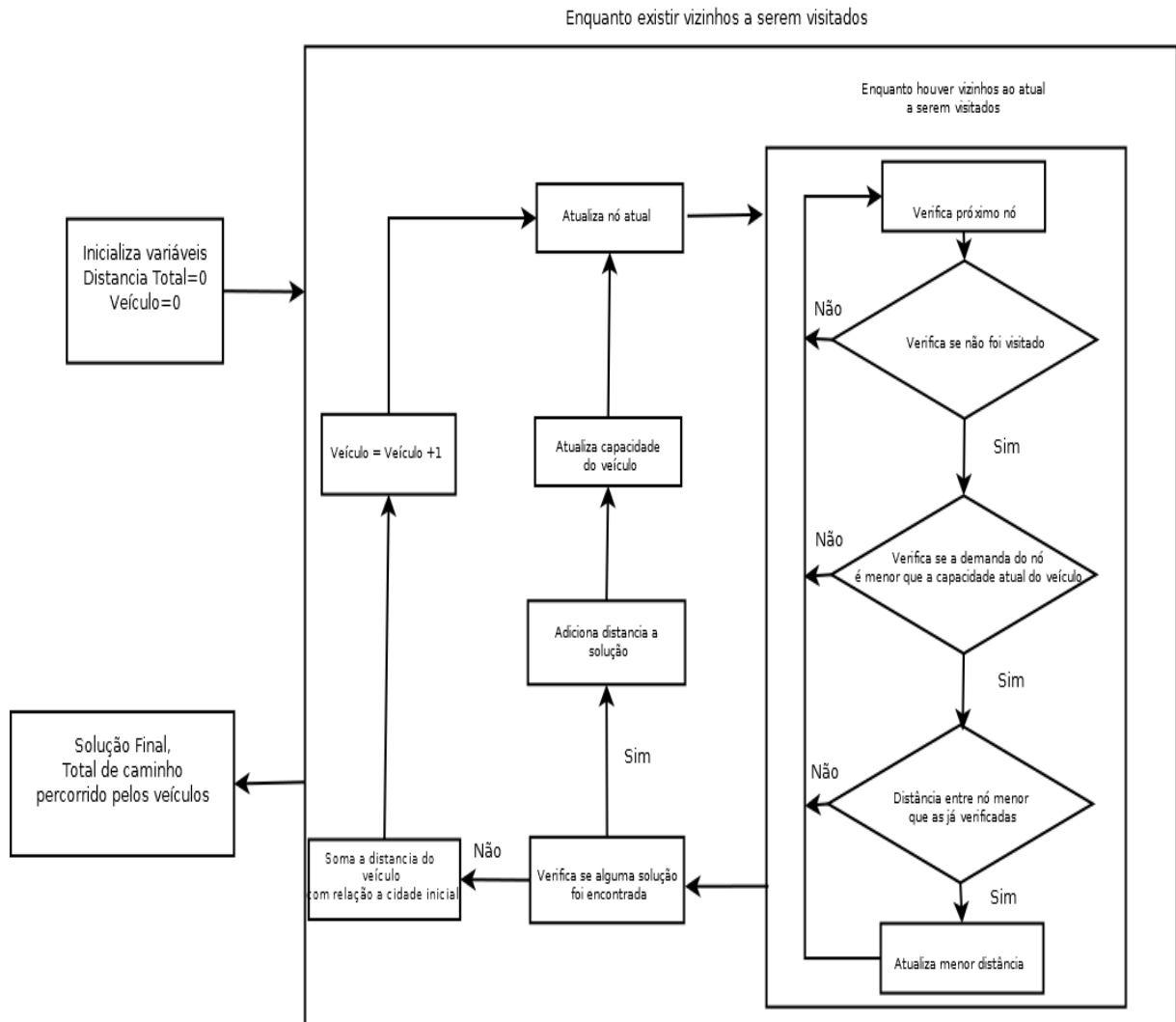


Figura 6 – Diagrama ilustrativo para o funcionamento do método guloso

3.2.2 COLÔNIA DE FORMIGAS

O funcionamento do algoritmo Colônia de Formigas segue de acordo com o pseudo código 4 apresentado :

Primeiramente são feitas as inserções do dados básicos para o algoritmo, como:

- Cidades

- Localização da cidade em coordenadas x e y em um plano 2D.

- Demanda da cidade especificada.
- Veículos
 - Localização do depósito a qual os veículos iniciarão sua trajetória.
 - Capacidade de cada veículo (serão todas iguais pois a frota é homogênea).

Com estes dados é possível obter a matriz de adjacências e realizar a inicialização do algoritmo (como visto na linha 1 do pseudo código), valores que são obtidos calculando a distância euclidiana 2D entre os dados de entrada x e y para cada nó. A linha dois se refere a inicialização da matriz de feromônios.

Considerando o algoritmo Colônia de Formigas, existem seis parâmetros configuráveis: $\alpha, \beta, \rho, \theta, t_{max}$ e t_{int} . Os parâmetros α e β tem influência na equação (1) para cálculo de probabilidades tendo a atratividade das cidades, sendo eles a influência dos traços de feromônio e das informações heurísticas, respectivamente]. O parâmetro ρ define a taxa de evaporação do feromônio em cada aresta. θ define a quantidade de cidades que poderão fazer parte da lista reduzida de adjacências. t_{max} e t_{int} são respectivamente o valor máximo de iterações e o valor máximo de iterações sem melhora para o algoritmo.

Após as etapas de inserção dos dados para geração da matriz e definição dos parâmetros o método entra em um *loop* no qual ficará pela quantidade de iterações máximas especificadas t_{max} ou quando atingir o parâmetro t_{int} de iterações sem melhora na resposta (Linha 3). Dentro do *loop* de iterações externo (Linha 3) é feito, para cada formiga, enquanto existirem cidades não visitadas por ela, os cálculos dos valores necessários para escolha da próxima cidade, só fazem parte do cálculo as cidades que não violem nenhuma das restrições para visita (Linha 6). Após o cálculo de valores, cada cidade terá um valor entre 0 e 1 diferente das demais. São selecionados então os θ de maior valor para comporem a lista de adjacência reduzida (Linha 11), da qual será selecionado um, de forma pseudo-aleatória, o qual então fará parte da solução, então a formiga muda seu estado para a cidade escolhida, marcando a posição atual como visitada e reduzindo da sua capacidade atual de acordo com o valor de demanda da cidade (Linha 13). Quando alguma formiga não encontra cidades que não violem nenhuma das restrições impostas mas, ainda existem cidades não visitadas, é julgado que um veículo terminou o seu percurso, com isso a cidade atual é alterada para a localização do depósito e é considerado que outro veículo iniciará seu percurso (Linha 8). Após todas as cidades serem visitadas, se da fim a geração de percurso da formiga em questão e outra solução tem inicio. Após todas as formigas gerarem suas soluções, acontece o depósito e evaporação de feromônio, no método Colônia de Formigas todas as formigas fazem o depósito de acordo com

a qualidade de sua solução, como apresentado na equação (2) , a evaporação acontece em todas as arestas de acordo com a equação (3), regida pelo parâmetro ρ (Linhas 18 e 20). Depois de todas as formigas gerarem suas soluções e os feromônios serem atualizados é verificado se alguma das soluções geradas pelas formigas é menor que a menor solução encontrada até a iteração passada, se sim, é atualizado o valor da melhor solução e t_{int} é zerado. Com o final da execução é exibido a melhor solução encontrada pelo algoritmo nesta execução.

Algorithm 4: Colônia de Formigas implementado pelo autor

```

1 Inicialização da matriz de adjacências;
2 Inicialização da matriz de feromônios;
3 while Critério de finalização não é satisfeito do
4   for Cada formiga do
5     while Existem cidades não visitadas do
6       Calcule o valor a ser atribuído a cada cidade que não viola nenhuma
7         restrição através da equação (1) ;
9       if Não existem cidades que não violam nenhuma restrição then
10        | Um veículo termina seu percurso e outro veículo inicia o seu;
11        end if
12        if Existem cidades que não violam nenhuma restrição then
13          | Selecione  $\Theta$  maiores valores para compor a lista de adjacências
14            reduzida;
15          | Selecione na lista de adjacências reduzida o próximo movimento
16            aleatoriamente;
17          | Reduza da capacidade do veículo a demanda da cidade escolhida;
18        end if
19      end while
20    end for
21  for Cada formiga do
22    | Atualização do feromônio para arestas visitadas;
23  end for
24  Evaporação;
25   $t_{max}++$ ;
26   $t_{int}++$ ;
27  if Melhor solução gerada nesta iteração < Melhor solução gerada até agora
28    then
29      | Melhor solução gerada até agora = Melhor solução gerada nesta iteração ;
30      |  $t_{int} = 0$ ;
31    end if
32 end while

```

3.2.3 COLÔNIA DE FORMIGAS ELITISTA

O funcionamento do algoritmo Colônia de Formigas Elitista segue de acordo com o pseudo código 5 a seguir:

O funcionamento do algoritmo 5 segue de maneira muito similar ao algoritmo 4 e seus parâmetros são os mesmos, exceto pela adição do Numero de Formigas Elitistas n_{fe} , o qual define a quantidade de formigas que serão autorizadas a realizar o deposito de feromônio nas arestas visitadas pelas mesmas, a diferença do algoritmo Colônia de Formigas apresentado pode ser visto nas linhas 20 e 21 do pseudo código, que é seleção das n_{fe} formigas de acordo com a qualidade da solução obtida na iteração atual e a restrição de n_{fe} formigas responsáveis pelo depósito de feromônio.

Algorithm 5: Colônia de Formigas Elitista implementado pelo autor

```

1 Inicialização da matriz de adjacências;
2 Inicialização da matriz de feromônios;
3 while Critério de finalização não é satisfeito do
4   for Cada formiga do
5     while Existem cidades não visitadas do
6       Calcule o valor a ser atribuído a cada cidade que não viola nenhuma
7         restrição através da equação (1) ;
8       if Nenhum valor foi calculado then
9         Um veículo termina seu percurso e outro veículo inicia o seu;
10      end if
11      if Existem valores calculados then
12        Selecione  $\Theta$  maiores valores para compor a lista de adjacências
13          reduzida;
14        Selecione na lista de adjacências reduzida o próximo movimento
15          aleatoriamente;
16        Reduza da capacidade do veículo a demanda da cidade escolhida;
17      end if
18    end while
19  end for
20  Pesquisa  $n_{fe}$  formigas com melhor solução nesta iteração;
21  for  $n_{fe}$  formigas com melhor solução do
22    Atualização do feromônio para arestas visitadas;
23  end for
24  Evaporação;
25   $t_{max}++$ ;
26   $t_{int}++$ ;
27  if Melhor solução gerada nesta iteração < Melhor solução gerada até agora
28    then
29    Melhor solução gerada até agora = Melhor solução gerada nesta iteração ;
30     $t_{int} = 0$ ;
31  end if
32 end while

```

3.3 TESTES E APRIMORAMENTOS

Foram realizados uma numerosa quantidade de testes paralelamente com o desenvolvimento da versão final do algoritmo. Nas primeiras versões, o numero de formigas utilizadas nas instâncias era uma constante. Posteriormente a análise das formas de implementação encontradas na literatura, foi adotado que a quantidade de formigas é sempre igual a quantidade de cidades na instância a ser executada. Também, nas primeiras versões não era utilizada a lista de adjacências reduzida, mas foi adicionado na versão final por demonstrar uma melhora no valor final da solução sem comprometer o tempo de execução.

3.4 DESCRIÇÃO DAS INSTÂNCIAS DE TESTE

Os testes computacionais foram realizados no conjunto de instâncias de Christofides et al. (1979), excluindo-se as que contêm restrição de tempo. O conjunto possui quatorze instâncias no total, sendo apenas sete utilizáveis com o algoritmo implementado, são elas as de número 1, 2, 3, 4, 5, 11 e 12, pois as demais possuem mais uma restrição, a limitação de tempo. As instâncias estão disponíveis para acesso no endereço <http://neo.lcc.uma.es/vrp/vrp-instances/capacitated-vrp-instances/> e foram escolhidas devido ao seu numeroso uso na literatura e a possibilidade de comparação com a melhor solução possível. A tabela 1 apresenta as instâncias utilizadas

Tabela 1 – Instâncias testadas

Instância	Clientes	Capacidade	Veículos usados	Solução Ótima
1	50	160	5	542.62
2	75	140	10	835.26
3	100	200	8	826.14
4	150	200	12	1028.42
5	199	200	17	1291.29
11	121	200	7	1042.11
12	101	200	10	819.56

A primeira coluna apresenta o número da instância definido no conjunto utilizado; a segunda coluna o número de clientes contidos na instância; a terceira coluna a capacidade dos veículos dada como restrição para este conjunto de dados ; a Coluna Quatro apresenta a quantidade de veículos mínima para completar o trajeto; a Coluna Cinco apresenta o resultado ótimos para soma das distâncias percorridas pelos veículos para cada instância.

No próximo capítulo são apresentadas os testes realizados e os resultados obtidos.

4 RESULTADOS

Este capítulo apresenta os resultados obtidos com uma série de variações nos parâmetros encontrados em cada método, posteriormente é comparado o resultado da configuração que forneceu os melhores valores com os encontrados na literatura.

4.1 AVALIAÇÃO DOS RESULTADOS ENCONTRADOS COM A VARIAÇÃO DOS PARÂMETROS

Os testes iniciais para escolha dos parâmetros foram realizados somente na instância 1, que é a instância com menor numero de cidades do conjunto. Consequentemente, leva-se um menor tempo para fornecer uma solução. Os parâmetros alterados para realização dos testes foram ρ , α , β e n_{fe} , mantendo os valores constantes de $\theta=5$, $t_{max}=1000$ e $t_{int}=40$. Para cada variação de parâmetro foram feitas 100 execuções do algoritmo. Os algoritmos implementados foram processados em um microcomputador com processador Intel(R) Core(TM) i5-2310 CPU@2.90Ghz e 8GB de memória RAM. Os resultados encontrado para o método Colônia de Formigas são apresentados nos figuras a seguir:

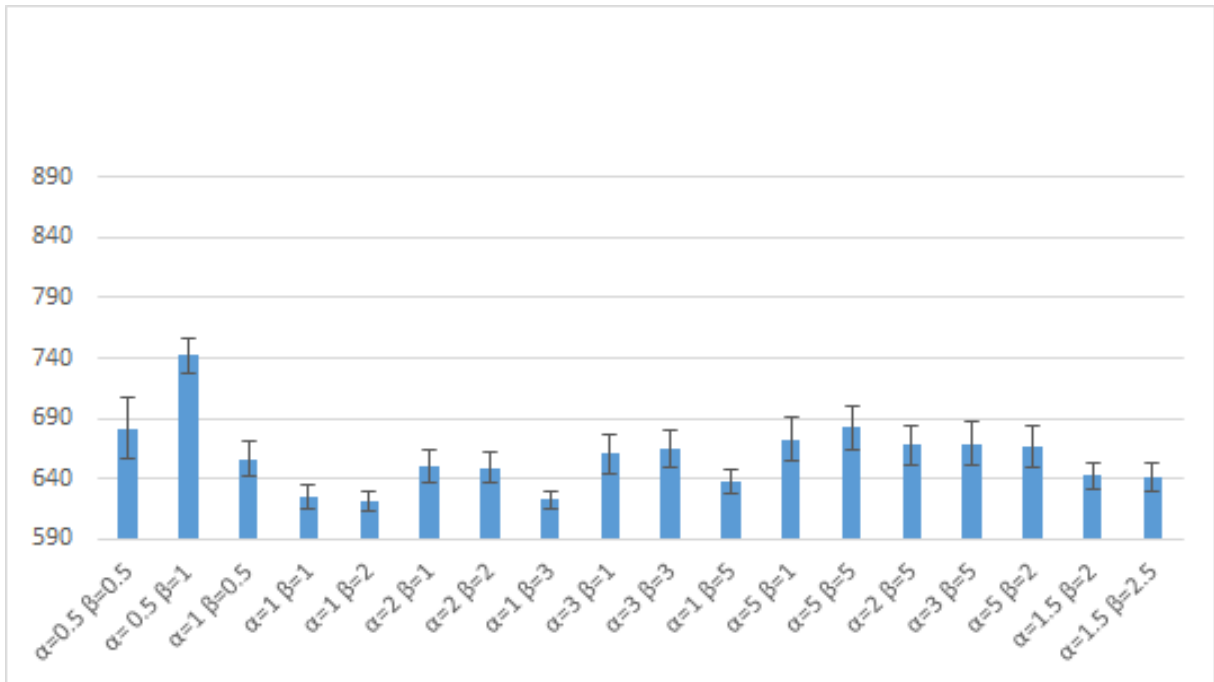


Figura 8 – Método Colônia de Formigas com variação dos parâmetros e $\rho = 0.2$

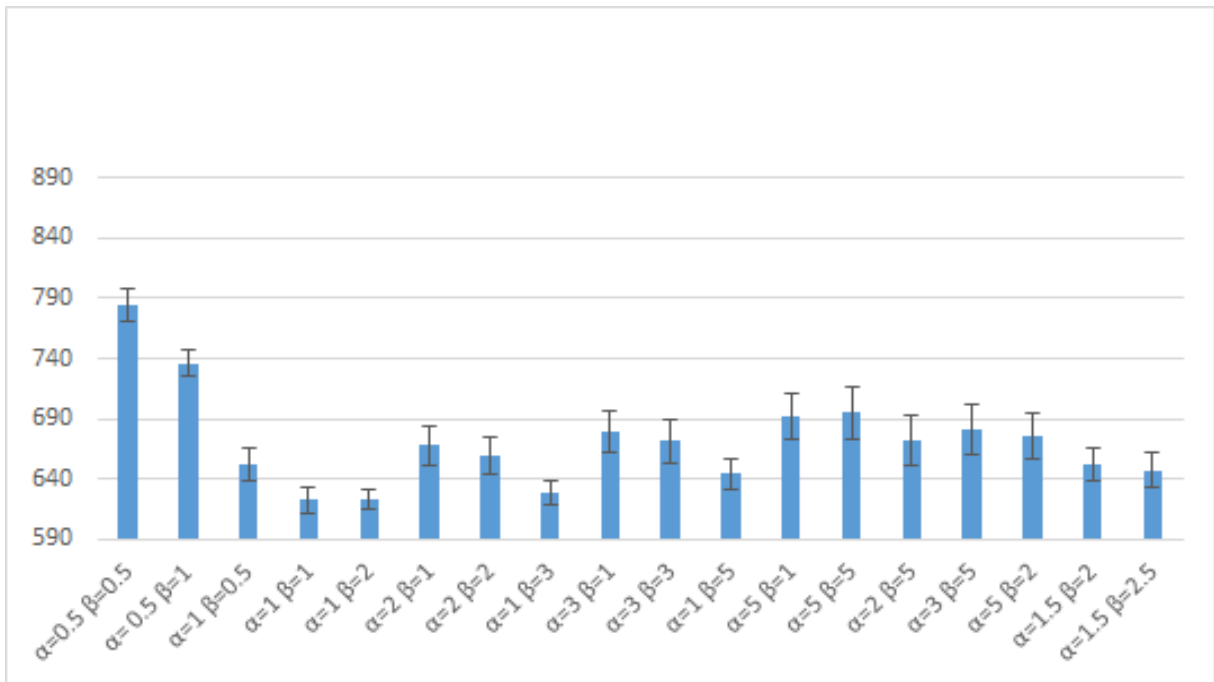


Figura 9 – Método Colônia de Formigas com variação dos parâmetros e $\rho = 0.5$

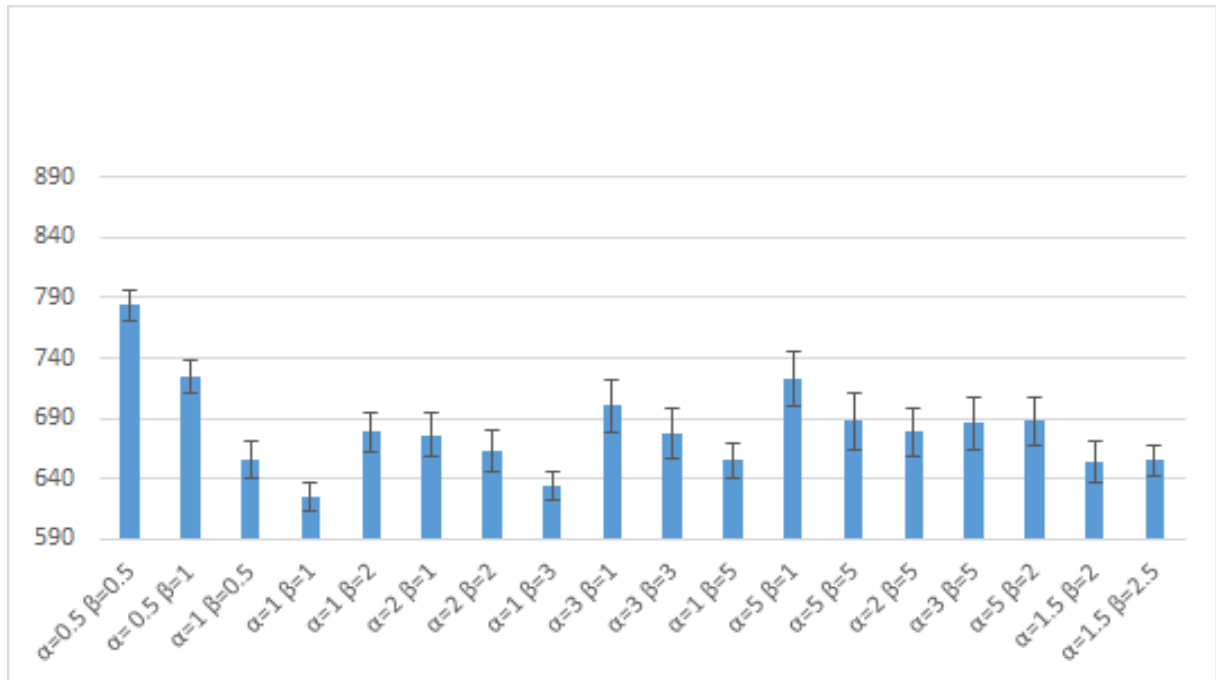


Figura 10 – Método Colônia de Formigas com variação dos parâmetros e $\rho = 0.8$

As Figuras 8, 9 e 10 representam os resultados obtidos com a variação dos parâmetros para evaporação $\rho = 0.2$, $\rho = 0.5$ e $\rho = 0.8$ respectivamente. Nas Figuras, o eixo vertical representa os valores de distância percorrida, o eixo horizontal apresenta a variação dos parâmetros realizada para obtenção dos resultados, cada coluna das figuras demonstra a média dos resultados obtidos para cada conjunto de parâmetros nas 100 execuções junto com seu desvio padrão.

Os resultados encontrados para o método Colônia de Formigas Elitista são apresentados a seguir:

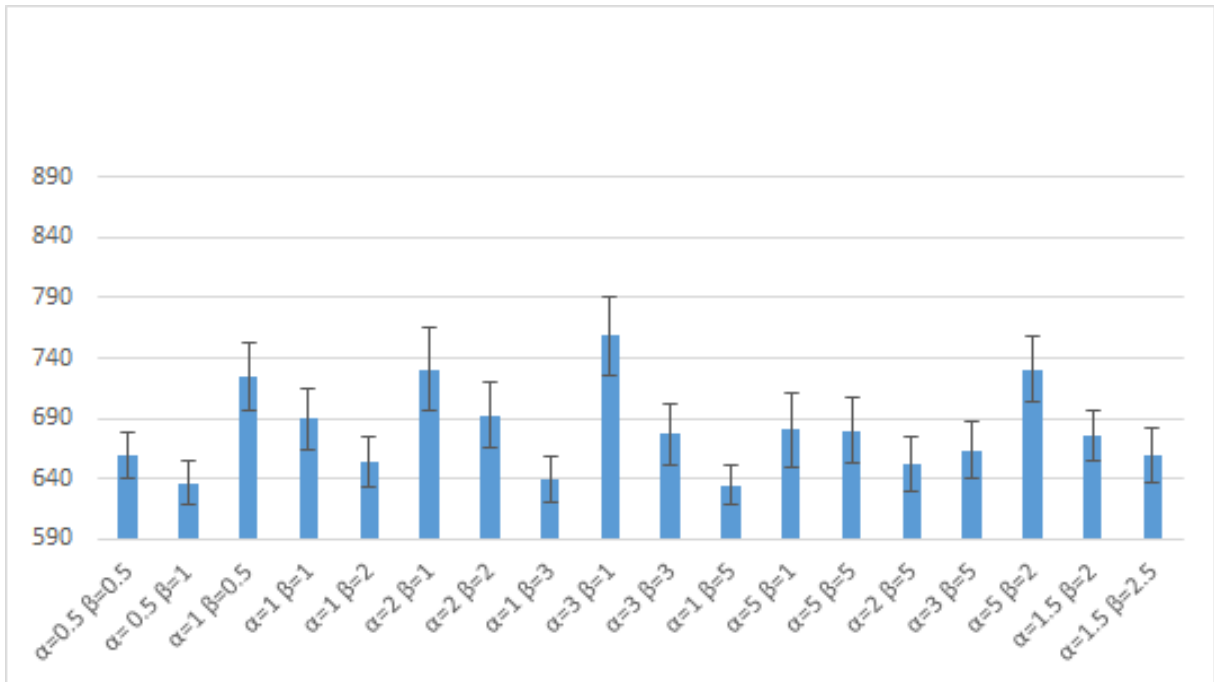


Figura 11 – Método Colônia de Formigas Elitista com variação dos parâmetros $\rho = 0.2$ e $n_{fe} = 2$

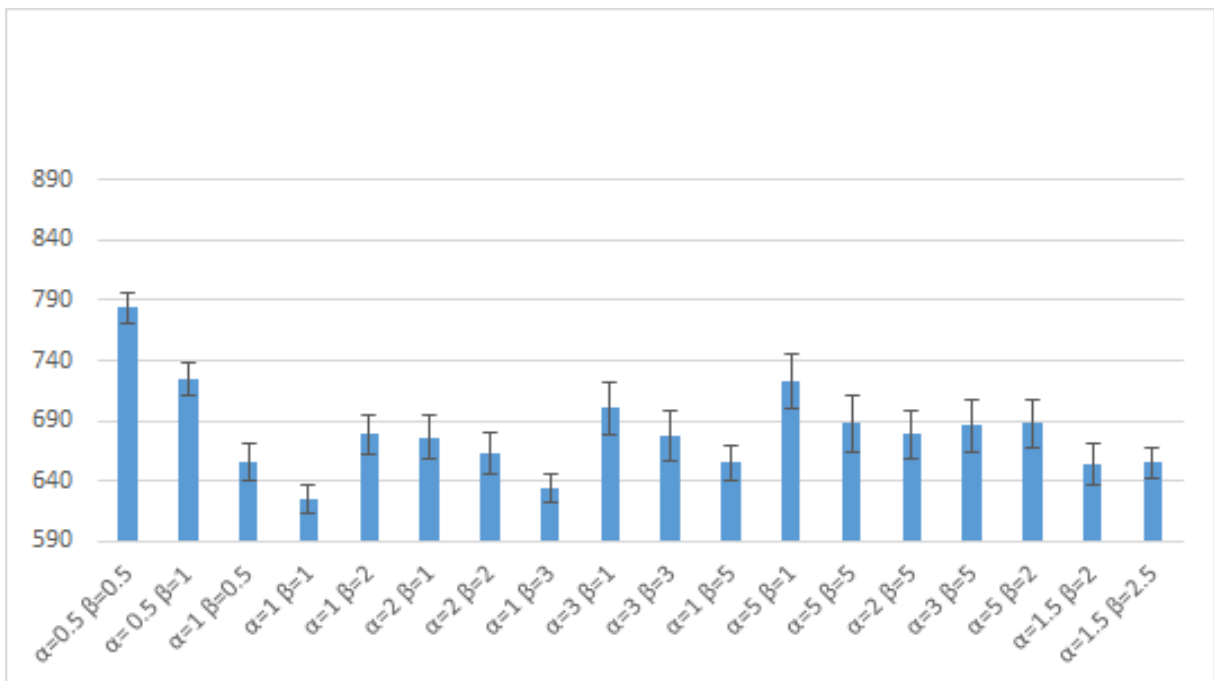


Figura 12 – Método Colônia de Formigas Elitista com variação dos parâmetros $\rho = 0.2$ e $n_{fe} = 5$

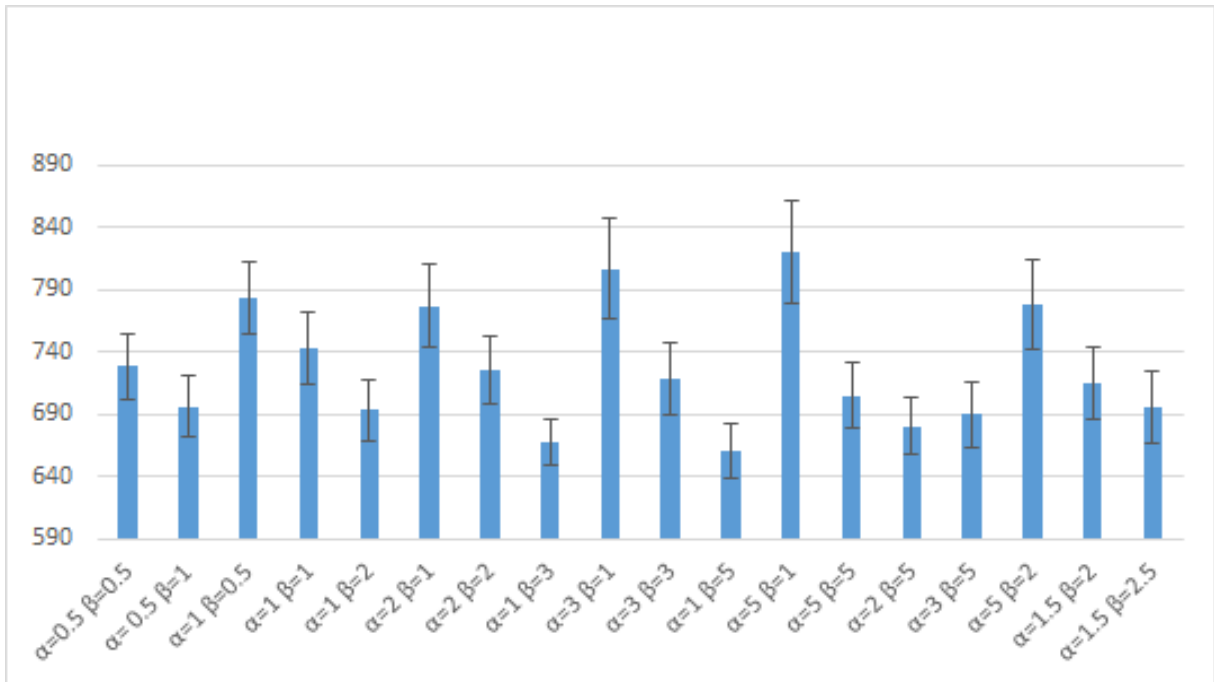


Figura 13 – Método Colônia de Formigas Elitista com variação dos parâmetros $\rho = 0.5$ e $n_{fe} = 2$

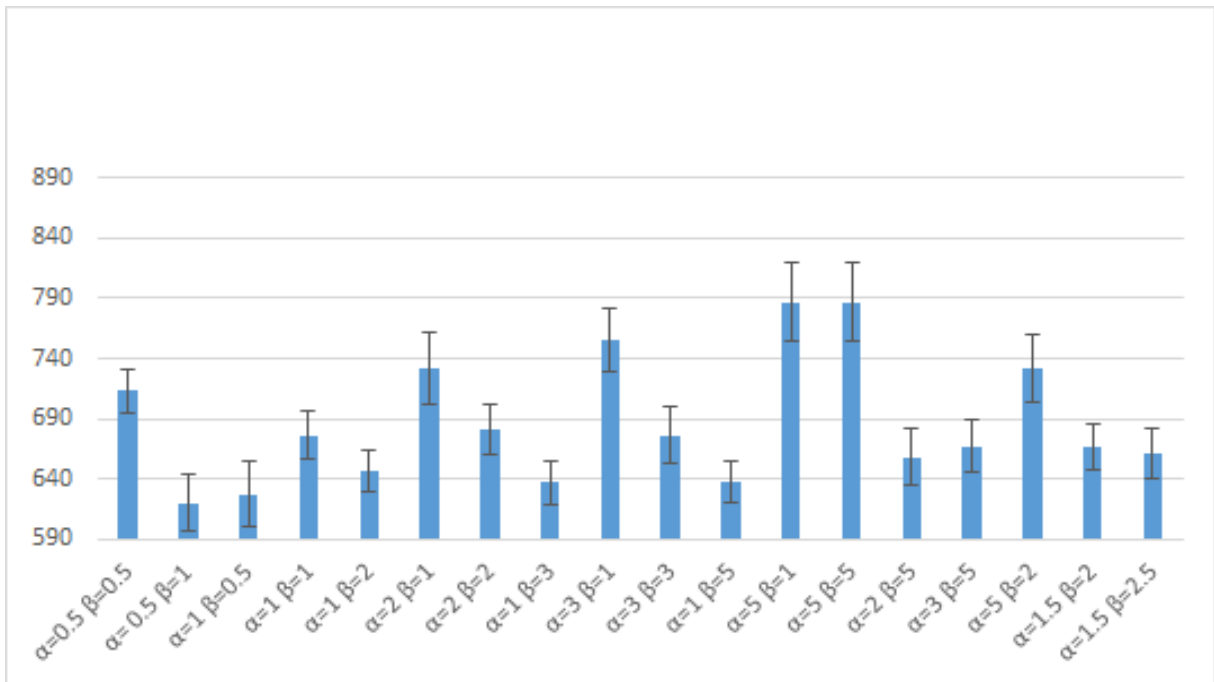


Figura 14 – Método Colônia de Formigas Elitista com variação dos parâmetros $\rho = 0.5$ e $n_{fe} = 5$

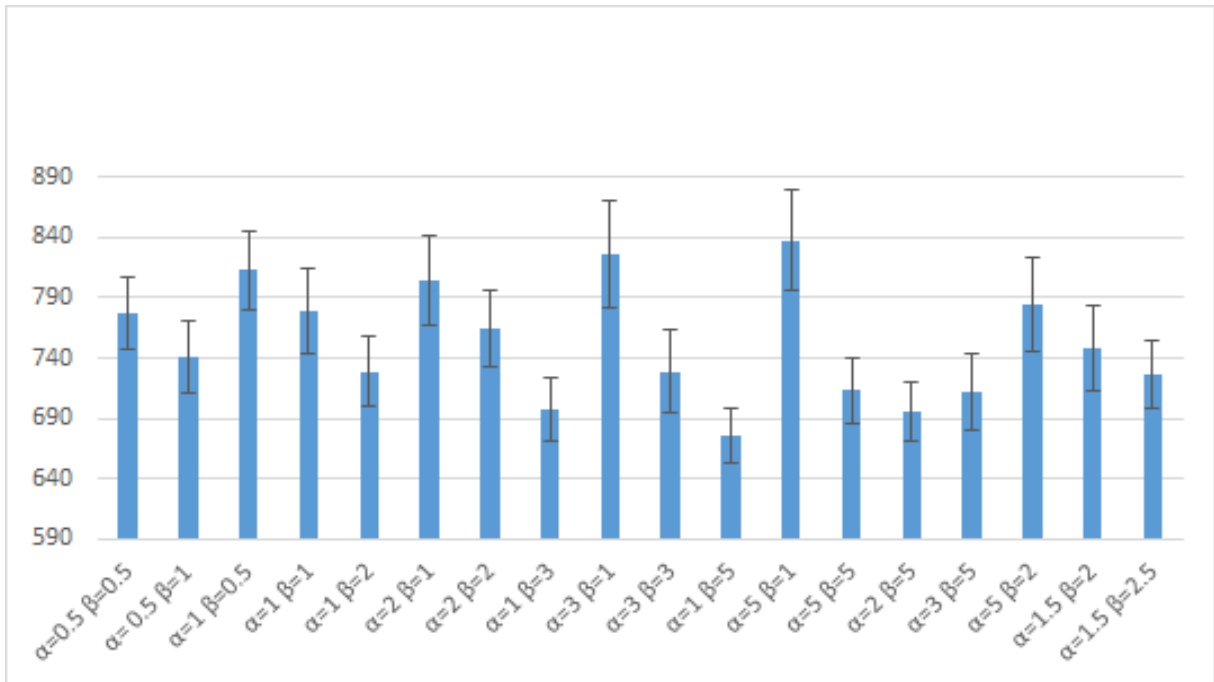


Figura 15 – Método Colônia de Formigas Elitista com variação dos parâmetros $\rho = 0.8$ e $n_{fe} = 2$

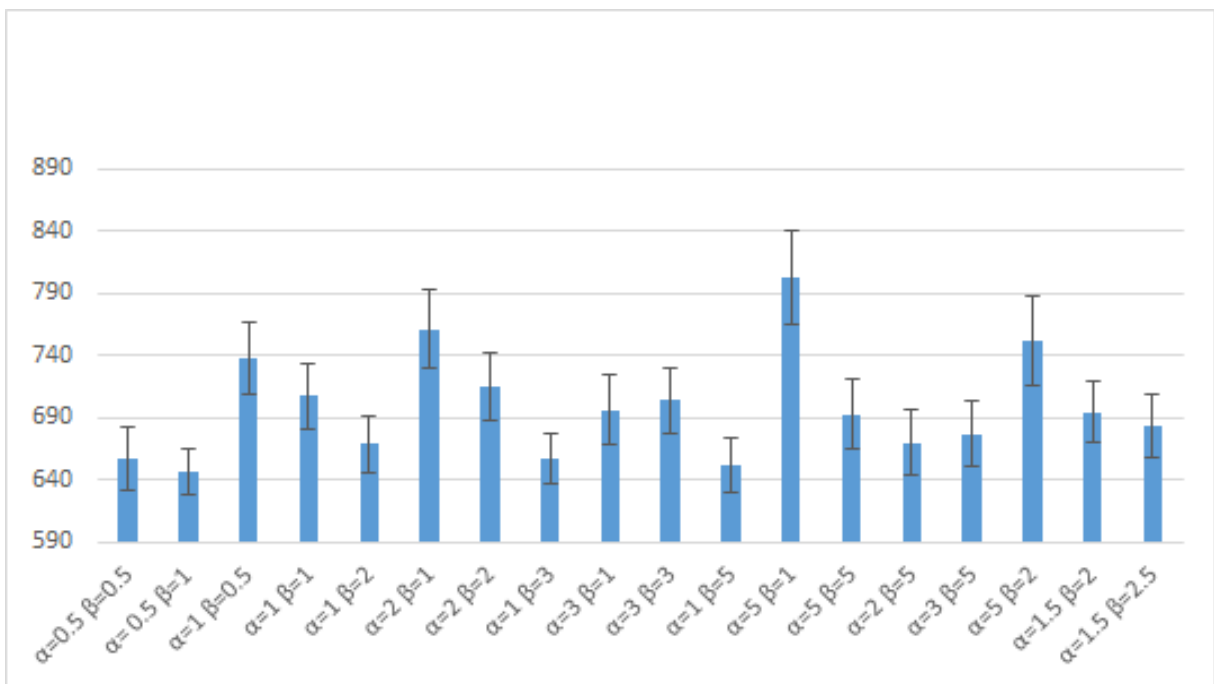


Figura 16 – Método Colônia de Formigas Elitista com variação dos parâmetros $\rho = 0.8$ e $n_{fe} = 5$

As Figuras 11, 12, 13, 14, 15 e 16 representam os resultados obtidos com a variação dos parâmetros para evaporação $\rho = 0.2$ com $n_{fe} = 2$ e $n_{fe} = 5$, $\rho = 0.5$ com $n_{fe} = 2$ e $n_{fe} = 5$ e $\rho = 0.8$ com $n_{fe} = 2$ e $n_{fe} = 5$, respectivamente. Nas Figuras, o eixo vertical representa os valores de distância percorrida, o eixo horizontal apresenta a variação dos parâmetros realizada para obtenção dos resultados, cada coluna das figuras demonstra a média dos resultados obtidos

para cada conjunto de parâmetros nas 100 execuções junto com seu desvio padrão.

Após avaliação dos resultados obtidos para a instância numero 1, foram escolhidos as melhores configurações, avaliando-se sua média e seu desvio padrão, para serem aplicados em testes nas demais instâncias, buscando-se a configuração de parâmetros que forneça os melhores resultados geral para elas. Essas configurações são:

- $\rho = 0,2 \quad \alpha = 1 \quad \beta = 1.$
- $\rho = 0,2 \quad \alpha = 1 \quad \beta = 2.$
- $\rho = 0,2 \quad \alpha = 1 \quad \beta = 3.$
- $\rho = 0,2 \quad n_{fe} = 5 \quad \alpha = 1 \quad \beta = 2.$
- $\rho = 0,2 \quad n_{fe} = 5 \quad \alpha = 1 \quad \beta = 3.$
- $\rho = 0,5 \quad \alpha = 1 \quad \beta = 1.$
- $\rho = 0,5 \quad n_{fe} = 5 \quad \alpha = 0,5 \quad \beta = 1.$
- $\rho = 0,5 \quad n_{fe} = 5 \quad \alpha = 1 \quad \beta = 0,5.$
- $\rho = 0,5 \quad n_{fe} = 5 \quad \alpha = 1 \quad \beta = 3.$
- $\rho = 0,8 \quad \alpha = 1 \quad \beta = 1.$

Os resultados obtidos aplicando-se estas configurações de parâmetros para as demais instâncias é apresentado a seguir:

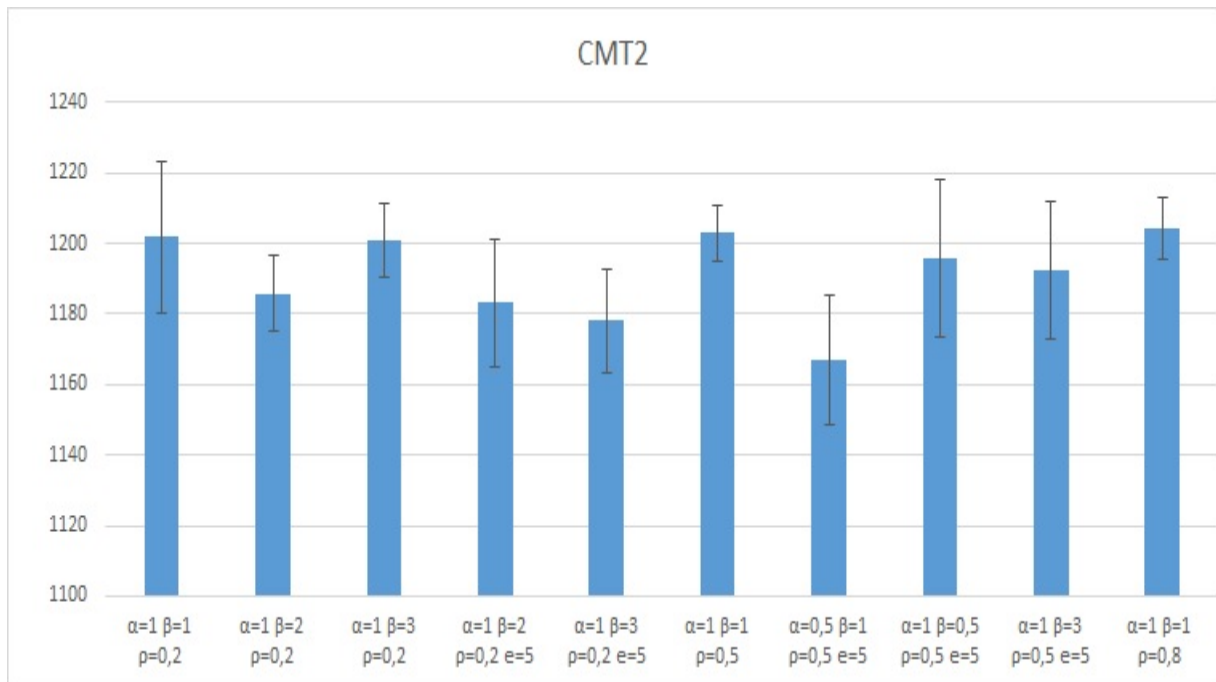


Figura 17 – Parâmetros aplicados na instância 2

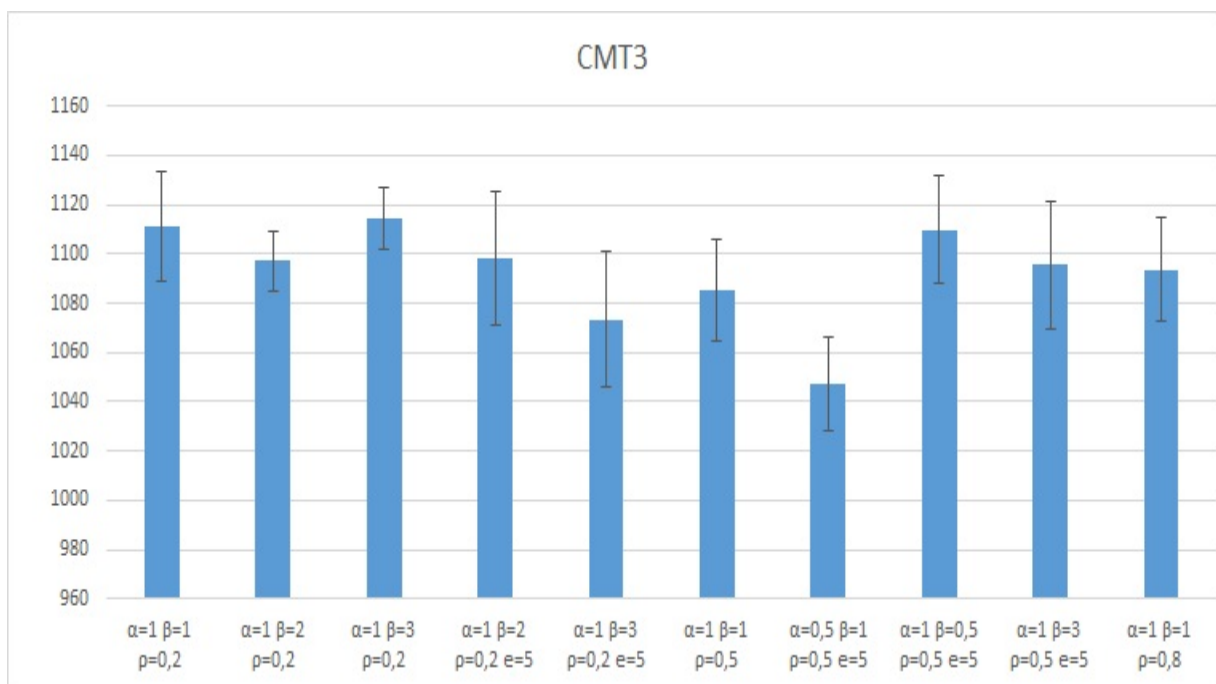


Figura 18 – Parâmetros aplicados na instância 3

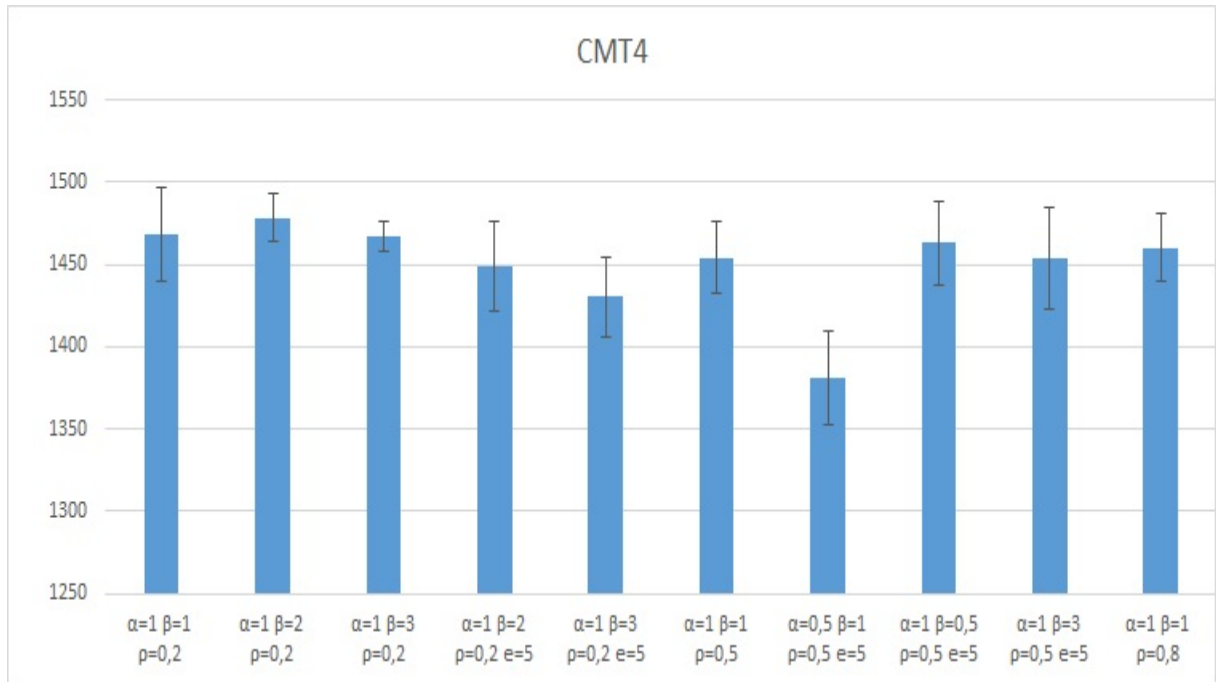


Figura 19 – Parâmetros aplicados na instância 4

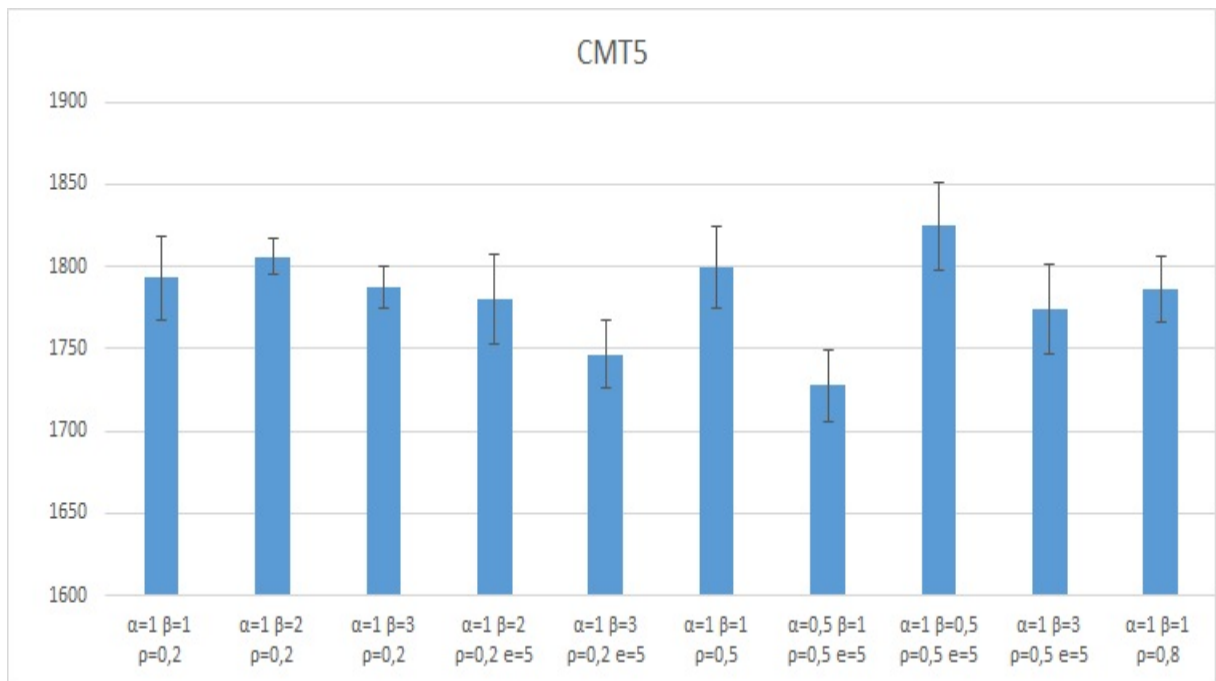


Figura 20 – Parâmetros aplicados na instância 5

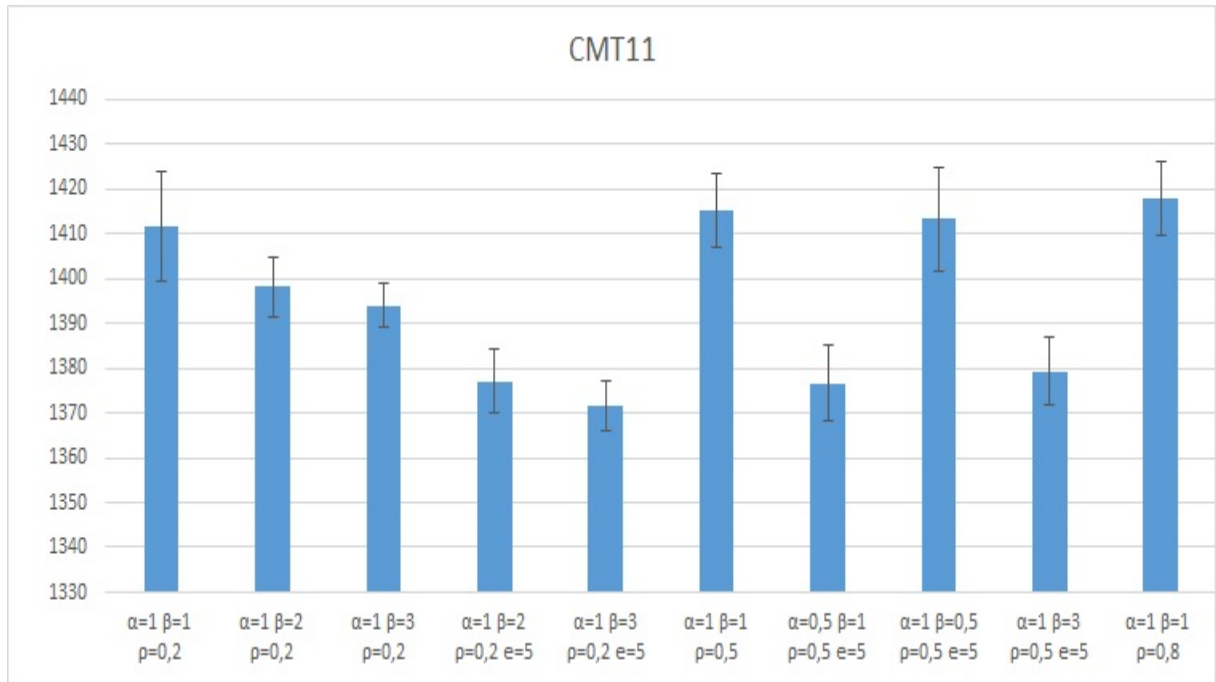


Figura 21 – Parâmetros aplicados na instância 11

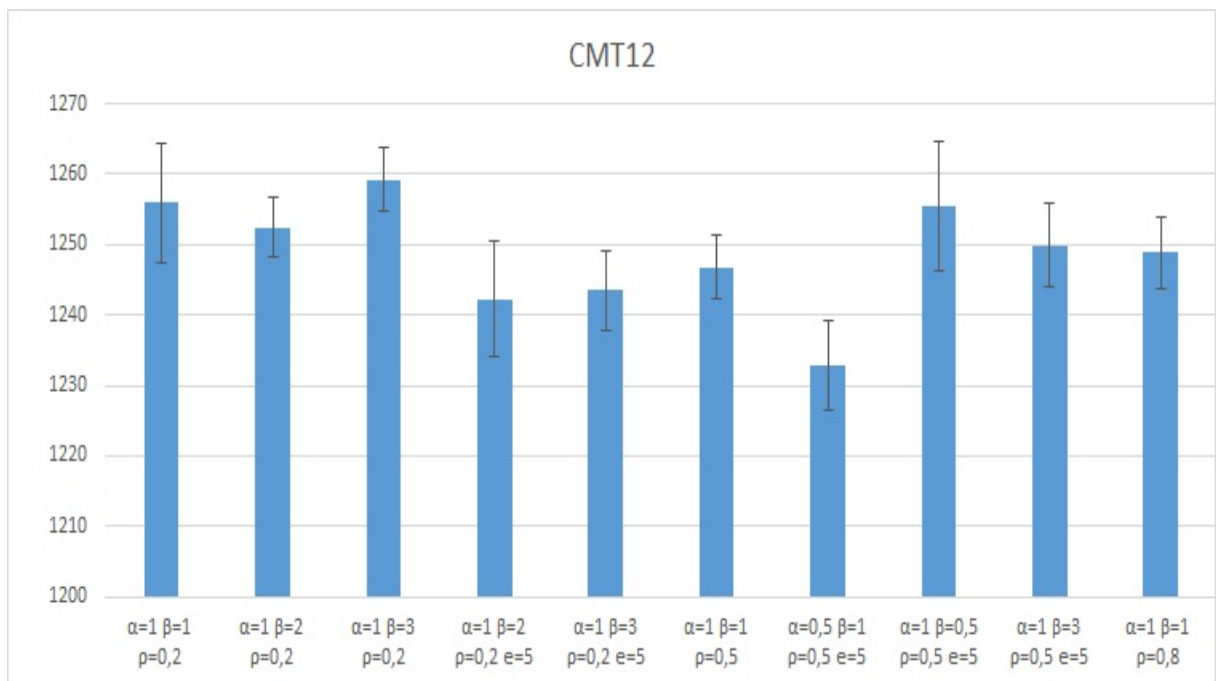


Figura 22 – Parâmetros aplicados na instância 12

As Figuras ??, 18, 19, 20, 21 e 22 apresentam os resultados obtidos aplicando os melhores parâmetros encontrados para a instância 1 nas instâncias 2, 3, 4, 5, 11 e 12. Nas Figuras, o eixo vertical representa os valores de distância percorrida, o eixo horizontal apresenta a variação dos parâmetros realizada para obtenção dos resultados, cada coluna das figuras demonstra a média dos resultados obtidos para cada conjunto de parâmetros nas 100 execuções

junto com seu desvio padrão.

4.2 COMPARAÇÃO DOS MELHORES RESULTADOS OBTIDOS COM OS DISPONÍVEIS NA LITERATURA

Após a fase de avaliação da qualidade das soluções obtidas com os parâmetros apresentados, foi escolhida a seguinte configuração para comparação de resultados com os encontrados na literatura:

- Colônia de Formigas
 - $\alpha = 1$.
 - $\beta = 1$.
 - $\rho = 0,5$.
 - Máximo de iterações = 1000.
 - Máximo de iterações sem melhora = 40.
 - Lista reduzida de adjacência = 5.

- Colônia de Formigas Elitista
 - $\alpha = 0,5$.
 - $\beta = 1$.
 - $\rho = 0,5$.
 - Máximo de iterações = 1000.
 - Máximo de iterações sem melhora = 40.
 - Lista reduzida de adjacência = 5.
 - Numero de formigas elitistas = 5.

Todas as variações foram executadas cem vezes para cada instância e seguem os parâmetros definidos anteriormente. Os resultados são apresentados na tabela 2.

Tabela 2 – Comparação dos Resultados

Inst.	Melhor	Guloso	ACO original				ACO Elitista				ACO+ 2opt		ACO+2opt+Swap	
			R	R_{dvp}	M_{dvp}	T_s	R	R_{dvp}	M_{dvp}	T_s	R_{dvp}	M_{dvp}	R_{dvp}	M_{dvp}
1	524.61	1085	586	11.70%	15.13%	0.48	582	10.9%	14.93%	0.54	0.00%	1.01%	0.00%	0.48%
2	835.26	2601	1173	40,43%	44,03%	1.32	1112	33,13%	39,72%	1.32	2.69%	3.53%	1.27%	2.12%
3	826.14	3160	1041	26,00%	31,13%	2.51	1002	21,28%	26,74%	2.83	3,19%	4.65%	1.47%	3.43%
4	1028.42	4932	1396	35,73%	41,40%	9.23	1335	29,81%	34,29%	8.72	7.44%	9.00%	4.94%	6.64%
5	1291.29	6177	1722	33,35%	39,35%	19.13	1662	28,70%	33,79%	18.53	10.65%	11.99%	8.17%	9.71%
11	1042.11	7127	1391	33,47%	35,79%	2.83	1356	30,12%	32,10%	2.77	1.83%	2.51%	1.31%	2.41%
12	819.56	3699	1235	50,69%	52,13%	1.92	1223	49,22%	50,04%	1.87	3.23%	5.06%	4.54%	5.35%

Os resultados apresentados seguem a seguinte configuração, na primeira coluna é apresentado o número N da instância testada, na coluna dois é apresentado o melhor valor possível para a instância. A coluna três contém os resultados encontrados para o algoritmo Guloso denotado por R_g . Na coluna quatro são apresentados os resultados encontrados para o algoritmo Colônia de Formigas o qual é composto por quatro valores, são eles R que é o melhor resultado encontrado para a instância, R_{dvp} que é o desvio padrão relativo da melhor solução encontrada e é gerido pela fórmula $DPV = \frac{Resultado\ obtido - Resposta\ otima}{Resposta\ otima} \times 100\%$, M_{dvp} é o desvio padrão relativo da média dos cem valores encontrados, T_s é o tempo médio de execução para as cem realizadas em segundos. Na coluna cinco se encontram os resultados encontrados para o algoritmo Colônia de Formigas Elitista, os quais seguem o mesmo padrão da coluna dois. Nas colunas seis e sete são apresentados os resultados encontrados em Tan et al. (2012), Colônia de Formigas Elitista com adição da busca local 2 – opt e Colônia de Formigas com adição das buscas locais 2 – opt e swap, ambas as colunas são compostas por R_{dvp} e M_{dvp} , que representam o desvio padrão relativo da melhor e da média das soluções encontradas respectivamente.

4.3 DISCUSSÃO SOBRE OS RESULTADOS OBTIDOS

Diante do apresentado, pode-se notar que a estratégia de otimização implementada atuou nos resultados de forma dependente a configuração dos parâmetros.

Como os parâmetros foram escolhidos através das realização de testes buscando a melhor configuração para a instâncias 1, o resultado de pior qualidade nas outras instâncias pode ser dado pela diferença nas dimensões e no modo como as cidades e demandas estão distribuídas nas demais.

Foram apresentados os resultados obtidos com as meta-heurísticas ACO e ACO Elitista, apenas com a adição da lista de adjacências reduzida. O objetivo inicial era comparar os resultados obtidos com meta-heurísticas não hibridizadas e sem adição de estratégias de busca local ou melhoramento de resultados, para analisar os desempenhos dos métodos implementados em um mesmo nível, porém, não foi encontrado na literatura implementações de meta-heurísticas voltadas a gerar soluções para o PRVC sem nenhuma dessas adições.

5 CONCLUSÕES

Os resultados obtidos com os métodos implementados demonstram que o ACO e sua variação, o ACO elitista, apresentam bons resultados quando comparados ao método guloso, porém, sem nenhuma hibridização com outros métodos, utilização de buscas locais ou heurísticas de melhoramento, como apresentadas em Tan et al. (2012), os algoritmos não são capazes de competir com as meta-heurísticas hibridizadas e com estratégias de melhoramento de solução disponíveis na literatura, porém, suas características de ser intuitivamente paralelizável, a possibilidade de execução com entradas dinâmicas e seu tempo médio com boa qualidade, fazem deles alternativas possíveis para gerar soluções dependendo do problema a ser resolvido.

5.1 TRABALHOS FUTUROS

Este trabalho pode ter sua continuidade voltada para o desenvolvimento de meta-heurísticas híbridas, aplicação de buscas locais, uso de heurísticas como estratégias de melhoramento ou a paralelização voltada aos algoritmos implementados. Também é possível a realização dos testes feitos na instância 1 em todas as outras instâncias, com objetivo de encontrar uma configuração geral de parâmetros que se adapte de melhor forma para todas as instâncias.

REFERÊNCIAS

- ASANOVIC, K. et al. A view of the parallel computing landscape. **Commun. ACM**, ACM, New York, NY, USA, v. 52, n. 10, p. 56–67, out. 2009. ISSN 0001-0782.
- BLUM, C.; ROLI, A. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. **ACM Comput. Surv.**, ACM, New York, NY, USA, v. 35, n. 3, p. 268–308, set. 2003. ISSN 0360-0300.
- BODIN, L. et al. Routing and scheduling of vehicles and crews : the state of the art. Pergamon Press, v. 10, n. 8, 1983.
- BONABEAU, E.; DORIGO, M.; THERAULAZ, G. **From Natural to Artificial Swarm Intelligence**. [S.l.]: Oxford University Press, 1999. ISBN 0195131584.
- CHRISTOFIDES, N.; MINGOZZI, A.; TOTH, P. **The vehicle routing problem**. [S.l.: s.n.], 1979.
- CLARKE, G.; WRIGHT, J. W. Scheduling of vehicles from a central depot to a number of delivery points. **Oper. Res.**, INFORMS, Institute for Operations Research and the Management Sciences (INFORMS), Linthicum, Maryland, USA, v. 12, n. 4, p. 568–581, ago. 1964. ISSN 0030-364X.
- COOK, W. J. et al. **Combinatorial Optimization**. 1997.
- CORMEN, T. et al. **Algoritmos: teoria e prática**. [S.l.]: CAMPUS - RJ, 2002. ISBN 9788535209266.
- CORMEN, T. et al. **Introduction to Algorithms**. [S.l.]: MIT Press, 2009. ISBN 9780262033848.
- DORIGO, M.; CARO, G. D. **New Ideas in Optimization**. Maidenhead, UK, England: McGraw-Hill Ltd., UK, 1999. 11–32 p.
- DORIGO, M.; STÜTZLE, T. **Ant Colony Optimization**. Scituate, MA, USA: Bradford Company, 2004. ISBN 0262042193.
- DROR, M.; LAPORTE, G.; TRUDEAU, P. Vehicle routing with split deliveries. **Discrete Applied Mathematics**, v. 50, n. 3, p. 239 – 254, 1994.
- GAREY, M.; JOHNSON, D. **Computers and Intractability: A Guide to the Theory of NP-completeness**. [S.l.]: W. H. Freeman, 1979. (Books in mathematical series). ISBN 9780716710448.
- GLOVER, F.; KOCHENBERGER, G. **Handbook of Metaheuristics**. [S.l.]: Springer US, 2006. (International Series in Operations Research & Management Science). ISBN 9780306480560.

GLOVER, F.; KOCHENBERGER, G. A. (Ed.). **Handbook of metaheuristics**. Boston, Dordrecht, London: Kluwer Academic Publishers, 2003. (International series in operations research & management science).

GOLDBARG, M.; LUNA, H. **Otimização combinatória e programação linear: modelos e algoritmos**. [S.l.]: Campus, 2000. ISBN 9788535205411.

GOODRICH, M. T.; TAMASSIA, R. **Algorithm Design and Applications**. 1st. ed. [S.l.]: Wiley Publishing, 2014. ISBN 1118335910, 9781118335918.

KIRKPATRICK C. D. GELATT, M. P. V. S. Optimization by simulated annealing. **Science**, American Association for the Advancement of Science, v. 220, n. 4598, p. 671–680, 1983.

KOLEN, A. W. J.; KAN, A. H. G. R.; TRIENEKENS, H. W. J. M. Vehicle routing with time windows. **Operations Research**, v. 35, n. 2, p. 266–273, 1987.

LADÁNYI, L.; RALPHS, T. K.; TROTTER JR., L. E. Branch, cut, and price: Sequential and parallel. In: **Computational Combinatorial Optimization, Optimal or Provably Near-Optimal Solutions [Based on a Spring School]**. London, UK, UK: Springer-Verlag, 2001. p. 223–260.

MICHALEWICZ, Z.; FOGEL, D. **How to Solve It: Modern Heuristics**. [S.l.]: Springer Berlin Heidelberg, 2013. ISBN 9783662078075.

NOVAES, A. G. **Logística e gerenciamento da cadeia de distribuição**. [S.l.: s.n.], 2007.

PEARL, J. **Heuristics: Intelligent Search Strategies for Computer Problem Solving**. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1984. ISBN 0-201-05594-5.

RESENDE, M.; RIBEIRO, C. **Greedy randomized adaptive search procedures**. [S.l.]: Kluwer Academic Publishers, 2003. 219–249 p.

SIPSER, M. **Introdução à teoria da computação**. [S.l.]: THOMSON PIONEIRA, 2007. ISBN 9788522104994.

STÜTZLE, T.; HOOS, H. H. Max-min ant system. **Future Gener. Comput. Syst.**, Elsevier Science Publishers B. V., Amsterdam, The Netherlands, The Netherlands, v. 16, n. 9, p. 889–914, jun. 2000. ISSN 0167-739X.

TAILLARD, . Parallel iterative search methods for vehicle routing problems. **Networks**, Wiley Subscription Services, Inc., A Wiley Company, v. 23, n. 8, p. 661–673, 1993. ISSN 1097-0037.

TALBI, E. **Metaheuristics: From Design to Implementation**. [S.l.]: Wiley, 2009. (Wiley Series on Parallel and Distributed Computing). ISBN 9780470496909.

TALBI, E.-G. **Metaheuristics: From Design to Implementation**. [S.l.]: Wiley Publishing, 2009. ISBN 0470278587, 9780470278581.

TAN, W. et al. Ant colony optimization for capacitated vehicle routing problem. v. 8, p. 846–852, 01 2012.

TOSCANI, L.; VELOSO, P. **Complexidade de Algoritmos: Série Livros Didáticos Informática UFRGS - Vol. 13**. [S.l.]: Bookman, 2009. ISBN 9788540701397.

TOTH, P.; VIGO, D. (Ed.). **The Vehicle Routing Problem**. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2001. ISBN 0-89871-498-2.