

**UNIVERSIDADE FEDERAL DE SANTA MARIA
CENTRO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM
ENGENHARIA DE PRODUÇÃO**

**ALGORITMO EVOLUTIVO PARA O PROBLEMA DO
CAIXEIRO VIAJANTE COM DEMANDAS
HETEROGÊNEAS**

DISSERTAÇÃO DE MESTRADO

Luis Eduardo Vieira

**Santa Maria, RS, Brasil
2006**

ALGORITMO EVOLUTIVO PARA O PROBLEMA DO CAIXEIRO VIAJANTE COM DEMANDAS HETEROGÊNEAS

por

Luis Eduardo Vieira

Dissertação apresentada ao Curso de Mestrado do Programa de Pós-Graduação em Engenharia de Produção, Área de Concentração em Tecnologia da Informação, da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para obtenção do grau de **Mestre em Engenharia de Produção.**

Orientador: Prof. Felipe Martins Müller

Santa Maria, RS, Brasil

2006

**Universidade Federal de Santa Maria
Centro de Tecnologia
Programa de Pós-Graduação em Engenharia de Produção**

A Comissão Examinadora, abaixo assinada,
aprova a Dissertação de Mestrado

**ALGORITMO EVOLUTIVO PARA O PROBLEMA DO CAIXEIRO
VIAJANTE COM DEMANDAS HETEROGÊNEAS**

elaborada por
Luis Eduardo Vieira

como requisito parcial para obtenção do grau de
Mestre em Engenharia de Produção

COMISSÃO EXAMINADORA:

Felipe Martins Müller, Dr. (UFSM)
(Presidente/Orientador)

José Vicente Canto dos Santos, Dr. (UNISINOS)

João Hélio Righi de Oliveira, Dr. (UFSM)

Santa Maria, 23 de novembro de 2006.

AGRADECIMENTOS

Primeiramente agradeço a Deus por ter chegado até aqui.

À minha família: Márcia, Débora e Melinna, que mais me incentivaram e compreenderam minha ausência em muitos momentos importantes, amo vocês.

Ao meu pai, sempre preocupado com minhas viagens, e também pelos momentos difíceis pelos quais ele passou.

À minha mãe, que por estar em outro plano, emanou forças para eu continuar essa caminhada.

Ao meu orientador: obrigado Felipe, principalmente pela exigência em fazer sempre o melhor, e isso acaba sendo a maior forma de incentivo.

Ao Marcus, pelas aulas de Java e pela ajuda na implementação dos programas.

À Suzi, que se não fosse sua insistência para que eu fizesse o processo seletivo não teria chegado até aqui hoje, valeu colega!

À URCAMP – Campus de São Gabriel, que muito me incentivou e colaborou para a conclusão da dissertação, principalmente aos professores Gerzson e Thadeu.

A todos os colegas de trabalho da URCAMP, que sempre torceram por mim.

Ao AMEM-CE UFSM nas pessoas dos professores Fábio Bastos e Felipe Müller, por disponibilizarem os recursos do laboratório para os testes dessa dissertação.

A todas as pessoas que de alguma forma contribuíram para a realização desse trabalho.

“É fazendo que se aprende a fazer aquilo
que se deve aprender a fazer.”

(Aristóteles)

RESUMO

Dissertação de Mestrado
Programa de Pós-Graduação em Engenharia de Produção
Universidade Federal de Santa Maria

ALGORITMO EVOLUTIVO PARA O PROBLEMA DO CAIXEIRO VIAJANTE COM DEMANDAS HETEROGÊNEAS

Autor: Luis Eduardo Vieira
Orientador: Felipe Martins Müller
Data e Local da Defesa: Santa Maria, 23 de novembro de 2006

O trabalho proposto nessa dissertação pertence à área de otimização combinatória, a qual visa encontrar uma solução para esses tipos de problema em um tempo computacional baixo e de forma eficaz. A otimização combinatória estuda um conjunto discreto de soluções, os quais possuem um número finito de elementos, para se poder encontrar a melhor solução viável para os problemas dessa grandeza. Uma das principais abordagens dessa área é o Problema do Caixeiro Viajante (PCV), principalmente devido à dimensão de possíveis soluções para o problema, fazendo com que seja intratável computacionalmente por métodos de buscas exaustivas. Face a todas essas características, este trabalho tem por objetivo estudar e desenvolver estratégias evolutivas para a resolução do Problema do Caixeiro Viajante com Demandas Heterogêneas (PCVDH), uma variação do PCV clássico. As estratégias evolutivas pertencem à classe da computação evolutiva, sendo métodos de busca inspirados na teoria da evolução das espécies, onde os melhores indivíduos competem pela sobrevivência. As estratégias evolutivas diferem das demais técnicas de otimização, pois a busca é realizada em uma população de soluções, não em um único ponto. Para a resolução do problema são propostos quatro algoritmos evolutivos, utilizando técnicas heurísticas e metaheurísticas para sua aplicação. Os resultados foram obtidos com testes utilizando instâncias de baixa densidade (baixa conexão), e comparados com a sua solução exata (solução ótima) e com outros métodos evolutivos encontrados na literatura. Esses resultados são avaliados com base na sua qualidade e tempo decorrido para sua execução.

Palavras-chave: Estratégia Evolutiva, Metaheurísticas, Otimização Combinatória, Problema do Caixeiro Viajante.

ABSTRACT

Master Thesis
Programa de Pós-Graduação em Engenharia de Produção
Universidade Federal de Santa Maria

ALGORITHM EVOLUTIONARY FOR THE TRAVELLING SALESMAN PROBLEM WITH HETEROGENEOUS DEMANDS

Author: Luis Eduardo Vieira
Advisor: Felipe Martins Müller
Santa Maria, november, 23 , 2006

The work proposed in this dissertation is the field of combinatorial optimization, which aims to find a solution to these types of problems at a low computational time and effectively. The combinatorial optimization studies a set of discrete solutions, which have a finite number of elements, to find the best viable solution to the problems of this magnitude. One of the main approaches that area is the Traveling Salesman Problem (TSP), mainly due to the size of possible solutions to the problem, so that is intractable computation by exhaustive search methods. Given all these features, this work is to study and develop evolutionary strategies for the resolution of the Problem of Traveling Salesman with Heterogeneous Demands (TSPHD), a variation of the classic TSP. The evolutionary strategies belong to the class of evolutionary computation, and methods of search based on the theory of the evolution of species, where the best individuals compete for survival. The evolutionary strategies differ from other optimization techniques, as the search is conducted in a population of solutions, not a single point. To solve the problem are proposed four evolutionary algorithms, using heuristics techniques and metaheurísticas for its implementation. The results were obtained from tests using instances of low density (low connection), and compared with the exact solution (optimal solution) and other progressive methods in the literature. These results are evaluated on the basis of their quality and time for its implementation.

Words-key: Evolutionary Strategy, Metaheuristics, Combinatorial Optimization, Traveling Salesman Problem.

LISTA DE FIGURAS

Figura 1- Construção de uma rota para o PCV	17
Figura 2- Componentes da classe dos Algoritmos Evolutivos.....	24
Figura 3- Algoritmo Evolutivo Básico.....	24
Figura 4- Geração de um novo indivíduo através da <i>evolution strategy</i> ES1	28
Figura 5- Regra de mutação da <i>evolution strategy</i> ES1	29
Figura 6- Exemplo de geração de um novo indivíduo através da <i>evolution strategy</i> ES2.....	30
Figura 7- Exemplo da mutação em um indivíduo	30
Figura 8- Exemplo de uma instância no PCVDH.....	33
Figura 9- Exemplo de conectividade dos grafos.....	34
Figura 10- Exemplo de grafo com arestas fantasmas	35
Figura 11- Matriz de cidades x cidades.....	36
Figura 12- Geração de uma rota inicial	36
Figura 13- Um movimento de troca 2-opt válido.....	38
Figura 14- Movimentos de trocas de arcos no algoritmo 3-opt.....	40
Figura 15- Movimentos de troca de um algoritmo or-opt.....	42
Figura 16- Geração de um novo indivíduo utilizando a Estratégia Evolutiva ES1	43
Figura 17- Passos da execução do algoritmo ES1.....	44
Figura 18- Geração de um novo indivíduo utilizando a Estratégia Evolutiva ES2	45
Figura 19- Execução do operador de recombinação PBX	46
Figura 20- Passos de execução do algoritmo ES2.....	46
Figura 21- Demonstração de uma troca swap	47

LISTA DE TABELAS

Tabela 1- Custos do ES1 para instâncias de 15 a 65 cidades [10 50]	51
Tabela 2- Custos do ES1 para instâncias de 15 a 65 cidades [20 10 100]	52
Tabela 3- Custos do ES1 para instâncias de 15 a 65 cidades [50 30 150]	54
Tabela 4- Tempo computacional para o ES1 com instâncias de 15 a 65 cidades ..	56
Tabela 5- Custos do ES2 para instâncias de 15 a 65 cidades [10 50]	58
Tabela 6- Custos do ES2 para instâncias de 15 a 65 cidades [20 10 100]	59
Tabela 7- Custos do ES2 para instâncias de 15 a 65 cidades [50 30 150].....	61
Tabela 8- Tempo computacional para o ES2 com instâncias de 15 a 65 cidades ..	63
Tabela 9- Custos do ES3 para instâncias de 15 a 65 cidades [10 50]	65
Tabela 10- Custos do ES3 para instâncias de 15 a 65 cidades [20 10 100]	66
Tabela 11- Custos do ES3 para instâncias de 15 a 65 cidades [50 30 150].....	68
Tabela 12- Tempo computacional para o ES3 com instâncias de 15 a 65 cidades ..	69
Tabela 13- Custos do ES4 para instâncias de 15 a 65 cidades [10 50]	71
Tabela 14: Custos do ES4 para instâncias de 15 a 65 cidades [20 10 100]	73
Tabela 15- Custos do ES4 para instâncias de 15 a 65 cidades [50 30 150].....	75
Tabela 16- Tempo computacional para o ES4 com instâncias de 15 a 65 cidades ..	77
Tabela 17- Custos para instâncias de 15 a 65 cidades [30 10 100]	79
Tabela 18- Tempo computacional para instâncias de 15 a 65 cidades [30 10 100]..	80

SUMÁRIO

1 INTRODUÇÃO	10
2 REVISÃO BIBLIOGRÁFICA	12
2.1 Otimização Combinatória	12
2.2 Conceitos sobre Teoria da Complexidade.....	13
2.3 Algoritmos Heurísticos.....	14
2.4 O Problema do Caixeiro Viajante	16
2.5 O Problema do Caixeiro Viajante com Demandas Heterogêneas.....	17
2.6 Trabalhos Correlatos.....	18
2.6.1 PCVDH – Método Exato.....	19
2.6.2 PCVDH – Métodos Heurísticos e Metaheurísticos	21
3 ALGORITMOS EVOLUTIVOS	23
3.1 Algoritmos Genéticos	25
3.2 Programação Evolutiva	25
3.3 Programação Genética.....	25
3.4 Estratégias Evolutivas	26
3.4.1 Teoria das Estratégias Evolutivas	26
3.4.2 Estratégias Evolutivas como Metaheurísticas	27
3.4.2.1 <i>Evolution Strategy</i> ES1	28
3.4.2.2 <i>Evolution Strategy</i> ES2.....	29
4 METODOLOGIA PARA A RESOLUÇÃO DO PROBLEMA	32
4.1 Características do PCVDH	32
4.2 Heurística do Vizinho Mais Próximo.....	35
4.3 Heurística de Melhoramento k-Optimal (k-opt).....	37
4.3.1 Heurística 2-Opt	37
4.3.2 Heurística 3-Opt	38
4.3.3 Heurística or-Opt	40
4.4 Teoria das Estratégias Evolutivas	42
4.4.1 Estratégia Evolutiva ES1	43
4.4.2 Estratégia Evolutiva ES2	45
4.4.3 Estratégia Evolutiva ES3	47
4.4.4 Estratégia Evolutiva ES4	47
5 RESULTADOS COMPUTACIONAIS	49
5.1 Algoritmo ES1	50
5.2 Algoritmo ES2	57
5.3 Algoritmo ES3	64
5.4 Algoritmo ES4	71
5.5 Algoritmos Evolutivos com Alta Densidade	78
6 CONCLUSÃO	81
7 REFERÊNCIAS BIBLIOGRÁFICAS	83

1 INTRODUÇÃO

Os problemas de otimização combinatória são definidos como um conjunto discreto de soluções, os quais possuem um número finito de elementos. Entretanto, alguns desses problemas não conseguem chegar a uma solução ótima em um tempo computacional baixo e de forma eficaz. Uma determinada solução pode ser considerada aceitável se estiver próxima da solução ótima.

Um dos problemas amplamente estudados em otimização combinatória é o Problema do Caixeiro Viajante (PCV), que consiste em encontrar o caminho de menor custo, dentro de um conjunto finito de cidades, partindo de uma cidade de origem, percorrendo as demais cidades uma única vez e retornando para a cidade de origem.

Esse problema tem uma descrição simples, porém a busca por uma boa solução não é uma tarefa das mais fáceis, pois a medida que aumenta o número de cidades, o problema cresce também em complexidade, dependendo da instância¹ que está sendo tratada.

Uma variação do PCV é o Problema do Caixeiro Viajante com Demandas Heterogêneas (PCVDH), que apresenta além de um custo fixo para se mover de uma cidade a outra, um custo variável que será calculado com base na demanda dos produtos a serem entregues em uma determinada cidade.

Existem na literatura alguns algoritmos que atacam esse problema (SARUBBI, 2003; PEREIRA; 2004). Particularmente neste trabalho é aplicado um algoritmo evolutivo para o PCVDH. Os algoritmos evolutivos são denominados algoritmos populacionais, pois ao invés de trabalhar com um único indivíduo na busca por uma solução, ele explora uma população de indivíduos (cada indivíduo representa uma possível solução do problema).

O principal objetivo desta dissertação é apresentar um algoritmo evolutivo para a resolução do Problema do Caixeiro Viajante Assimétrico com Demandas Heterogêneas e avaliar os resultados apresentados com base nas instâncias criadas por Sarubbi (2003) e Pereira (2004).

¹ dados de entrada dos problemas que servem para testes do algoritmo

O presente trabalho está dividido em seis capítulos. O Capítulo I é onde está inserida esta parte introdutória. No Capítulo II é apresentado o PCV, bem como sua formulação matemática. Nesse capítulo é tratado também o PCVDH, algoritmos heurísticos e um breve conceito de otimização combinatória.

O Capítulo III descreve os algoritmos evolutivos, com suas principais características. O Capítulo IV é dedicado aos métodos de implementação do algoritmo evolutivo para o PCVDH. No Capítulo V são apresentados e analisados os resultados computacionais obtidos da aplicação do algoritmo evolutivo. Por fim, no Capítulo VI, são apresentadas as conclusões sobre o trabalho, suas contribuições e recomendações para trabalhos futuros.

2 REVISÃO BIBLIOGRÁFICA

Este capítulo apresenta alguns conceitos e um breve histórico sobre otimização combinatória, Teoria da Complexidade, o problema do caixeiro viajante, e do objeto de estudo dessa dissertação que é o problema do caixeiro viajante com demandas heterogêneas, cujo objetivo é de minimizar o custo de um caixeiro no seu deslocamento entre um conjunto de cidades, partindo da cidade de origem, passando por todas as cidades uma única vez e retornando a cidade de origem. O tempo de execução do algoritmo também é um parâmetro importante na execução de uma instância.

2.1 Otimização Combinatória

Como já foi citado na Introdução desse trabalho, um Problema de Otimização Combinatória(POC) contém um conjunto finito de possíveis soluções. Um POC tem também como característica a otimização de uma função objetivo, que deve obedecer o conjunto de restrições impostas pelo problema. Se a solução encontrada não violar as restrições do problema, então temos uma solução factível. Uma solução que violar pelo menos uma das restrições será considerada uma solução infactível. Por exemplo, uma solução factível para o PCVDH deve admitir que cada vértice tenha somente uma aresta de entrada e uma aresta de saída. Os problemas de otimização combinatória podem ser de minimização (custos) ou maximização (lucros). Se o problema for de minimização devemos encontrar uma solução para o problema de forma que o valor da função objetivo seja o menor possível. Quando o problema for de maximização, esse valor deverá ser o maior possível (GOLDBARG; LUNA, 2000). Deve-se considerar também, que, encontrada uma solução factível e essa solução é a melhor possível, chegamos então à solução ótima do problema.

Para a resolução de problemas de otimização combinatória utilizamos algoritmos exatos e algoritmos heurísticos. Se o algoritmo encontra a solução ótima para o problema, isto é, o valor da função objetivo é garantido como o melhor que qualquer outro valor encontrado, então temos um algoritmo exato. Na prática ocorre a existência de alguns problemas cujo número de soluções cresce de forma exponencial na medida em que se aumenta o tamanho do problema – no caso do

PCVDH aumenta o número de vértices (cidades) - na Teoria da Complexidade (GAREY; JOHNSON, 1979 e CAMPELLO; MACULAN, 1994) esses problemas pertencem a classe tipo **NP** (GAREY, 1979). Garey afirma ainda que no caso dos algoritmos exatos encontrarem a solução em tempo polinomial, esse problema pertence então a classe tipo **P**. Os algoritmos exatos não analisam todo o espaço de soluções do problema, pois eles trabalham sob uma abordagem fundamentada matematicamente, que, em um espaço S de soluções, algumas características do problema indicam que determinadas soluções não podem alcançar o valor ótimo. No PCVDH, como veremos na seção **XXXX**, o método exato pára se encontrar um vértice que não tenha um par de arestas como entrada e saída (estará violando uma restrição do problema). Nos trabalhos de Bodin et al. (1983) e Laporte et al. (1992) são apresentados maiores detalhes sobre os métodos exatos.

Se um algoritmo não garante encontrar a solução ótima, mas uma solução viável, isto é, que se aproxime da solução ótima com um tempo computacional aceitável, e cujas instâncias do problema sejam maiores que as instâncias do algoritmo exato, então ele é chamado de algoritmo heurístico ou aproximativo.

2.2 Conceitos sobre Teoria da Complexidade

Nesta seção serão apresentados breves conceitos sobre a Teoria da Complexidade. Como foi dito na seção anterior, os problemas que são resolvidos por um algoritmo exato (determinístico) em tempo polinomial pertencem a classe de problemas denominada classe **P** (***Polinomial Time***). Em problemas para os quais não é conhecido um algoritmo que encontre uma solução em tempo polinomial, diz-se pertencer a classe **NP** (***NonDeterministic Polinomial Time***), sendo considerados de difícil resolução. Dessa forma, pode-se afirmar que $P \subseteq NP$, mas por outro lado não sabemos se $P = NP$; para se provar isso, todos os problemas pertencentes à classe NP deveriam ser resolvidos em tempo polinomial, ou então serem reduzidos polinomialmente para os problemas da classe P.

Problemas da classe **NP-difícil** caracterizam-se por haver redução polinomial a partir de todo problema pertencente à classe NP. A classe **NP-completo** é composta por problemas pertencentes à interseção das classes NP e NP-difícil. Pode-se dizer que um problema de decisão $D \in NP$ é NP-completo se todos os outros problemas de NP se transformam polinomialmente em D . Então, se existir um

algoritmo polinomial para a resolução de algum problema NP-completo, todos os problemas da classe NP também poderão ser resolvidos em tempo polinomial. Dessa forma, será possível mostrar que $P = NP$. Até os dias de hoje ainda não foi possível encontrar tal problema. Como já foram realizadas muitas pesquisas sobre essa questão, há fortes evidências que isso nunca será possível.

O PCV pertence a classe NP-completo. Pode-se citar, entre outros problemas pertencentes a essa classe, o problema da mochila, coloração de grafos, programação inteira e problema de Steiner em grafos.

2.3 Algoritmos Heurísticos

As heurísticas são procedimentos criados para resolver um problema através do conhecimento, sendo aplicáveis a um determinado tipo de problema, com suas características particulares que serão analisadas e exploradas a fim de se obter uma boa solução. Portanto, uma heurística não possui, geralmente, uma formulação matemática padrão que possa ser direcionada a problemas genéricos. Os algoritmos heurísticos são divididos, nessa dissertação, em construtivos e de melhoramento (OSMAN, 1991).

Algoritmos construtivos são aqueles que começam com uma solução vazia e passo a passo vão agregando componentes (tarefas, arestas, vértices, variáveis, entre outras) à solução para o problema, de acordo com as especificações do mesmo, e somente ao final das iterações do algoritmo se obtém uma solução. Esta é considerada como a solução inicial do problema, e torna-se um limitante para a procura por novas soluções, porque, por exemplo, se o problema for de minimização, uma solução com um valor de função objetivo associado maior que a solução inicial, então essa solução não é promissora. Se o problema for de maximização, uma solução com um valor de função objetivo associado menor que a solução inicial, então essa solução também não é promissora.

Algoritmos de melhoramento partem de uma solução inicial factível, normalmente encontrada através de uma heurística de construção, realizando uma busca local através de movimentos de trocas e inserções, com o objetivo de encontrar uma solução de melhor qualidade. O algoritmo pára a sua execução quando um critério de parada foi atingido, ou seja, a partir da solução corrente não se encontra uma solução melhor com as tentativas do algoritmo, sendo essa

chamada de ótimo local (MÜLLER, 1993). Esta heurística também é encontrada na literatura com a descrição de heurística de busca local ou de vizinhança.

Se uma solução pode sofrer alterações sob alguma técnica heurística, então ela possui um número limitado de novas configurações as quais ela pode, imediatamente, assumir. O conjunto das novas soluções (melhores ou não) que podem ser obtidas imediatamente a partir da aplicação de alguma técnica sobre uma solução é denominado vizinhança daquela solução, e é determinado de acordo com a configuração da solução antes da aplicação da técnica. Algoritmos que chegam em um ponto onde a vizinhança de alguma solução não produz melhoria e então, param, são chamados de heurísticas de Busca Local, e o resultado obtido por esta heurística é chamada de ótimo local.

Um ótimo local pode se tornar uma armadilha para a heurística que, não podendo realizar nenhuma melhoria na solução, pode estar em um ponto afastado da solução ótima do problema. A solução ótima, em comparação ao ótimo local, é denominada ótimo global.

Para escapar das armadilhas impostas pelos ótimos locais e evitar a ciclagem do algoritmo, foram desenvolvidas outras técnicas chamadas de metaheurísticas.

A metaheurística é um processo iterativo ou de refinamento de solução de problema que organiza e direciona heurísticas subordinadas, pela combinação de diferentes conceitos, podendo manipular uma solução completa, incompleta ou um conjunto de soluções, tentando evitar parada prematura em ótimo local através de mecanismos que permitem escapar do mesmo, permitindo inclusive uma deterioração controlada de soluções para diversificar a busca. Tendo como objetivo explorar características de boas soluções e até novas regiões promissoras, saindo de um ótimo local, esta procura é guiada pelo algoritmo de busca local.

As buscas utilizadas em metaheurísticas podem ser divididas em duas categorias: uma delas compreende os métodos que exploram várias soluções (vizinhanças) a cada iteração (algoritmos populacionais) e a outra categoria compreende os métodos que exploram apenas um elemento de uma vizinhança a cada iteração.

Os algoritmos populacionais exploram uma população de soluções a cada iteração, sendo esta estratégia de busca capaz de explorar várias regiões do espaço de soluções a cada execução. Assim, não se constrói uma única trajetória de busca, pois as novas buscas são combinações das soluções anteriores (BURIOL, 2000).

Nesta categoria, atualmente existem várias metaheurísticas, dentre as mais conhecidas pode-se citar os algoritmos genéticos (Goldberg, 1989), algoritmos meméticos (MOSCATO, 1989, 1999), Scatter Search (GLOVER, 1990), Ant Colony Systems (STÜTZLE; DORIGO, 1999) e as estratégias evolutivas que foram inicialmente criadas para problemas de otimização no caso contínuo (RECHENBERG, 1973) e atualmente estão sendo aplicados para problemas de otimização discreta (HOMBERGER; GEHRING, 1999).

Algoritmos com apenas um elemento da vizinhança a cada iteração, geram um caminho ou trajetória de soluções, obtidos pela transição de uma solução a outra de acordo com os movimentos permitidos pela metaheurística. Como exemplo desta classificação temos Simulated Annealing (KIRKPATRICK et al., 1983), Busca Tabu (GLOVER; LAGUNA, 1993), GRASP (Greedy Randomized Adaptive Procedure) (FEO; RESENDE, 1994), Redes Neurais (POTVIN, 1993) e Simulated Jumping (AMIN, 1999).

2.4 O Problema do Caixeiro Viajante

O Problema do Caixeiro Viajante – PCV (CHRISTOFIDES, 1979) é um dos problemas de otimização combinatória que tem uma vasta aplicabilidade, e devido a isso, há décadas esse problema vem sendo estudado, surgindo assim centenas de trabalhos com novas idéias e alternativas para encontrar melhores soluções para o problema (em termos de tempo computacional e qualidade da solução).

O PCV é definido como segue: dado um grafo não direcionado $G=(V,E)$, onde V é um conjunto de n vértices e E é um conjunto de pares (i, j) para cada aresta que liga o vértice i ao vértice j ($i, j = 1, \dots, n$), sendo $(i, j) \in E$, é possível encontrar um ciclo Hamiltoniano de G , conforme a figura 1.

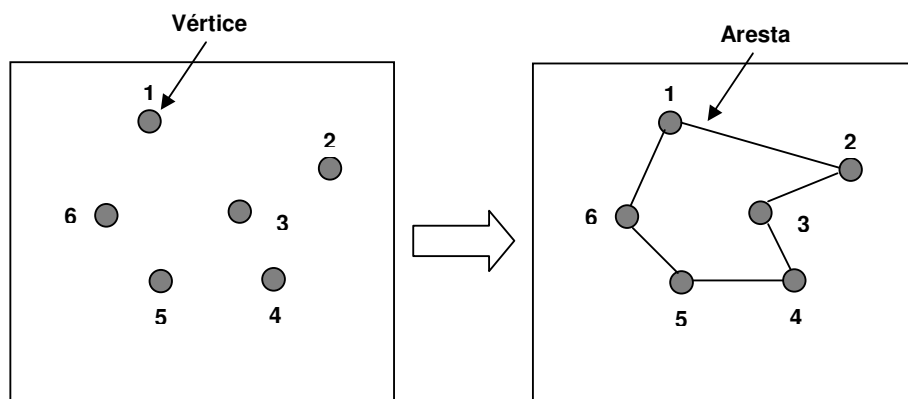


Figura 1 – Construção de uma rota para o PCV

Esse problema possui algumas características, a saber: dada uma instância para o PCV pode-se tratá-la como sendo totalmente conectada, isto é, para cada par de vértices i e j pertencentes a V existe uma aresta correspondente (i, j) pertencente a E ; em contra-partida se existir pelo menos um par de vértices que não estejam conectados, consideramos então que essa instância não possui 100% de conectividade. Para a existência de um ciclo Hamiltoniano não se necessita de uma instância 100% conectada. Para cada aresta (i, j) existe um valor c_{ij} agregado, o qual representa o custo do deslocamento do vértice i ao vértice j .

O objetivo desse problema é encontrar uma rota representando um Ciclo Hamiltoniano de menor custo. A função objetivo pode ser matematicamente descrita como:

$$\text{Min} \sum_{(i, j) \in R} C_{ij} \quad , \text{ onde } R \text{ é um subconjunto de } E, \text{ o}$$

conjunto de arestas (i, j) pertencentes à rota de solução.

Quando o custo do PCV para se deslocar do vértice i para o vértice j for o mesmo para se deslocar do vértice j para o vértice i , O PCV é considerado simétrico, caso contrário é assimétrico.

2.5 O Problema do Caixeiro Viajante com Demandas Heterogêneas

Por herança este problema apresenta as características do PCV e outras características de operacionalização para frota de veículos. Conforme Sarubbi (2003), o problema básico é focado no qual uma frota de veículos com capacidade fixa (número, capacidade de armazenamento, velocidade e outras) é utilizada para o transporte de mercadorias ou pessoas entre localidades de entrega e coleta, sendo

que cada localidade possui uma demanda associada. Dessa forma o problema é determinar que demandas serão transportadas e qual rota será seguida por cada veículo para um determinado conjunto de demandas. Sarubbi afirma ainda que o objetivo mais comumente buscado é minimizar o custo total com combustível, pessoal e depreciação dos veículos. Devem ser observadas também as restrições do problema, como capacidade de transporte, janela de tempo e outras.

O Problema do Caixeiro Viajante com Demandas Heterogêneas (PCVDH) trata de um acréscimo de demandas aos nós ou cidades que farão parte do grafo, analisando então o comportamento e o custo da rede com a inclusão dessas demandas diferenciadas. A formulação do problema consiste de um grafo $G = (V, E)$ onde V representa o conjunto dos nós e E dos arcos. Ainda temos um nó de origem O , local de partida do caixeiro e um conjunto K que é o complemento de O em relação a V , e que para cada nó $k \in K$ existe uma demanda heterogênea d_k . Esta característica pode servir na modelagem de problemas que avaliam o grau de prioridade de entrega dos produtos.

A função objetivo é:

$$\text{Min} \sum_{(i,j) \in E} (b_{i,j} x_{i,j} + \sum_{k \in K} c_{i,j,k} f_{i,j,k})$$

onde,

x_{ij} , será 1 se o caixeiro passa pelo arco (i,j) ; 0 caso contrário.

b_{ij} é o custo fixo para percorrer o arco (i,j) ;

c_{ijk} é o custo variável para transportar uma unidade de produto k através do arco (i,j) ;

f_{ijk} é o fluxo que passa pelo arco (i,j) destinado ao nó de demanda k ;

Este trabalho pretende obter um custo menor em relação ao trabalho de PEREIRA (2004), o qual atacou o mesmo problema. Para isso serão utilizadas também as mesmas instâncias de Sarubbi (SARUBBI, 2003).

2.6 Trabalhos correlatos

Não existem até o momento na literatura, muitos trabalhos relacionados ao PCVDH. Por isso será feita uma breve análise dos métodos de Sarubbi (2003) e de

Pereira (2004), os quais abordaram sobre o mesmo problema. Cabe salientar também que será dada maior ênfase no método de Sarubbi, pois esse trabalho utiliza as mesmas instâncias e formulação matemática do seu método exato.

2.6.1 PCVDH – Método Exato

Sarubbi e Pacca (SARUBBI, 2003) definiram este tipo de problema como Problema do Caixeiro Viajante com Demandas Heterogêneas, e propuseram a seguinte modelagem: considerando um grafo $G = (V, E)$ onde V representa os nós e E são as arestas que identificam os pares de nós entre os quais o caixeiro pode passar. Existe um nó de origem o , que é o local de saída do caixeiro viajante, e um conjunto $K \subseteq V$, que é o complemento de o em relação a V , e que para cada nó $k \in K$ existe uma demanda heterogênea d_k que deve ser atendida. Como existe um custo fixo associado à utilização de qualquer arco e um custo variável proporcional ao fluxo no arco, o problema é determinar um ciclo que minimize a somatória de custos fixos de utilização e dos custos variáveis dos custos dos arcos.

Sarubbi definiu para esse problema uma formulação de programação linear, descrita da seguinte forma:

$$x_{ij} = \begin{cases} 1 & \text{se o caixeiro passa pelo arco } (i,j) \\ 0 & \text{caso contrário} \end{cases}$$

- b_{ij} é o custo fixo para percorrer o arco (i,j) com o caminhão vazio;
- c_{ijk} é o custo variável para transportar uma unidade de produto k através do arco (i,j) ;
- f_{ijk} é o fluxo que passa pelo arco (i,j) destinado ao nó de demanda k .

A modelagem matemática ficaria:

$$\text{Min} \sum_{(i,j) \in E} (b_{i,j} x_{i,j} + \sum_{k \in K} c_{i,j,k} f_{i,j,k}) \quad 2.1$$

sujeito às seguintes restrições:

$$- \sum f_{ojk} = - d_k \quad \forall k \in K \quad 2.2$$

$$(o,j) \in E$$

$$\sum_{(i,k) \in E} f_{ijk} = d_k \quad \forall k \in K \quad 2.3$$

$$\sum_{(i,k) \in E} f_{ijk} - \sum_{(j,l) \in E} f_{jlk} = 0 \quad \forall k \in K, j \neq k \quad 2.4$$

$$f_{ijk} \leq d_k x_{ij} \quad \forall (i,j) \in E, \forall k \in K \quad 2.5$$

$$f_{ijk} \geq 0 \quad \forall (i,j) \in E, \forall k \in K \quad 2.6$$

$$\sum_{(i,j) \in E} x_{ij} = 1 \quad \forall j \in V \quad 2.7$$

$$\sum_{(i,k) \in E} x_{ij} = 1 \quad \forall i \in V \quad 2.8$$

$$x_{ij} \in \{0, 1\} \quad \forall (i,j) \in E \quad 2.9$$

$$-\sum_{(o,j) \in E} g_{oj} = -\sum_{k \in K} d_k \quad \forall k \in K \quad 2.10$$

$$\sum_{(i,k) \in E} g_{ik} - \sum_{(k,j) \in E} g_{kj} = d_k \quad \forall (i,j) \in E \quad 2.11$$

$$g_{ij} \leq \left(\sum_{k \in K} d_k x_{ij} \right) \quad \forall (i,j) \in E \quad 2.12$$

$$g_{ij} \geq 0 \quad \forall (i,j) \in E \quad 2.13$$

Como foi citado anteriormente, a função objetivo 2.1 considera não somente os custos fixos, mas também os custos variáveis referentes aos fluxos dos produtos. A restrição 2.2 assegura que o fluxo que sai do nó de origem é igual ao valor da soma das demandas de todos os nós de demanda. A restrição 2.3 determina que a demanda de um produto k é igual ao fluxo desse produto que chega ao vértice k . A restrição 2.4 assegura a conservação do fluxo de um dado produto através dos vértices de passagem daquele produto. A restrição 2.5 faz com que o fluxo nos arcos não escolhidos para a rota solução seja nulo. A não negatividade do fluxo de cada produto k é assegurada na restrição 2.6. A restrição 2.7 garante que incide um, e somente um, arco em cada nó, e a restrição 2.8 assegura que sai um, e somente um, arco a partir de cada nó. A restrição 2.9 constitui que as variáveis x_{ij} são binárias. A restrição 2.10, relativa ao fluxo global, afirma que o fluxo total que sai do nó de origem é igual ao somatório das demandas dos destinos dos diversos produtos. A restrição 2.11, que também é relativa ao fluxo global, considera que o fluxo proveniente de um determinado nó menos o que sai é igual a demanda deste nó. A restrição 2.12 assegura que, se existe fluxo em uma aresta, o valor de g_{ij}

desta aresta é menor ou igual ao somatório de todas as demandas. Caso contrário, o valor de g_{ij} é igual a zero. A última restrição, 2.13, assegura a não negatividade das variáveis g_{ij} .

Foram encontrados resultados para instâncias (ver explicação na seção 4.1) de até 65 vértices, utilizando para isso o *software* Cplex 7.0 e uma resolução através do método de decomposição de Benders (GEOFFRION; GRAVES, 1974). Sarubbi utilizou como limitante para a aplicação dos algoritmos o tempo de execução dos mesmos.

2.6.2 PCVDH – Métodos Heurísticos e Metaheurísticos

O trabalho desenvolvido por Pereira (2004) aborda o mesmo problema dessa dissertação, pois são utilizadas heurísticas e metaheurísticas para a resolução do PCVDH. Foram desenvolvidos os algoritmos evolutivos ES1 e ES2 (seções 3.4.2.1 e 3.4.2.2, respectivamente).

Conforme Pereira, além do grafo G de arestas e vértices e do valor de custo fixo c_{ij} para cada aresta do grafo, existe um valor de demanda d_{ijk} associado a cada vértice k , representando uma demanda maior ou menor para o produto que será entregue na cidade k . É definida também a cidade de origem. Por fim, existe para cada nó k um custo variável b_{ijk} , que representa o custo para transportar uma unidade do produto com destino ao nó k através da aresta (i, j) . A Função Objetivo para encontrar o menor custo ficaria:

$$\text{Min} \sum_{(i,j) \in E} (c_{ij} + \sum_{k \in K} b_{ijk} d_{ijk})$$

O algoritmo para o cálculo do custo será demonstrado a seguir, com base na seguinte especificação: seja S a representação de uma solução, um vetor cujos valores s_i ($i = 1, \dots, n$) representam o índice de qual vértice está na i -ésima posição na rota de solução a partir da origem (a_1 deve ser o índice do vértice de origem e a_n o índice do último vértice da rota), v_i determina se o vértice i foi ou não visitado no cálculo do custo. No início do algoritmo, para qualquer i , v_i possui o valor NÃO_VISITADO. Abaixo a estrutura do algoritmo: (PEREIRA, 2004)

- 1) Faça $p = 1$; custo = 0, $i = s_i$;
- 2) Faça $v_i = \text{VISITADO}$; $p = p + 1$; $i = s_p$; $j = s_{p+1}$;
- 3) Faça custo = custo + c_{ij} ;
- 4) Para k de 1 até n faça:
 - 4.1) Se v_k for NÃO_VISITADO, então faça custo = custo + $d_k b_{ij}$;
- 5) Se p for menor que n , então vá para o **passo 2**, senão faça $j = p_1$;
- 6) Faça custo = custo + c_{ij} ;

PARE, a variável custo possui o valor de Função Objetivo para a solução representada por S.

Neste capítulo foi feita uma revisão bibliográfica referente à otimização combinatória, teoria da complexidade, heurísticas, problema do caixeiro viajante clássico, PCVDH e ainda trabalhos relacionados ao tema dessa dissertação. No próximo capítulo serão abordados os principais algoritmos evolutivos.

3 ALGORITMOS EVOLUTIVOS

Existe uma ordem de problemas em otimização combinatória que possui um alto grau de complexidade, praticamente inviabilizando o uso de algoritmos exatos que poderiam ser aplicados na sua resolução (LEWIS; PAPADIMITRIOU, 1981). Essa dificuldade em resolver problemas com essas características utilizando métodos exatos deve-se também ao rigor dos modelos matemáticos. Para dirimir todas essas dificuldades surgiram as heurísticas e metaheurísticas, e dentro deste contexto os Algoritmos Evolutivos.

Algoritmos Evolutivos são métodos computacionais baseados em uma analogia com a evolução natural, e pertencem a classe da Computação Evolutiva, inspiradas na computação biológica. Outro exemplo de computação biológica são as redes neurais (POTVIN, 1993), que fazem uma analogia com o sistema nervoso do ser humano.

Na aplicação do algoritmo evolutivo para a resolução de um problema específico, são feitas uma variação randômica e posterior seleção de indivíduos em uma população de soluções candidatas, e estes indivíduos regularmente representam soluções potenciais para o problema. Mecanismos de recombinação e mutação também são utilizados nessa estrutura com a finalidade de gerar novos indivíduos, que são diferentes da estrutura existente. A evolução simulada incluirá algum tipo de método de seleção onde os indivíduos mais aptos ao ambiente terão maior probabilidade de sobreviver, de acordo com uma função de adaptabilidade ou *fitness*.

Alguns conceitos básicos sobre os principais componentes pertencentes à classe dos algoritmos evolutivos, exemplificado na figura 2:

- Indivíduo – também chamado de cromossomo, é um membro da população, uma estrutura de dados representando uma possível solução para um determinado problema;
- Gene – representa uma determinada posição da estrutura de dados;
- Alelo – valores que podem ser assumidos por um gene;
- Locus – posição ocupada por um gene no indivíduo;

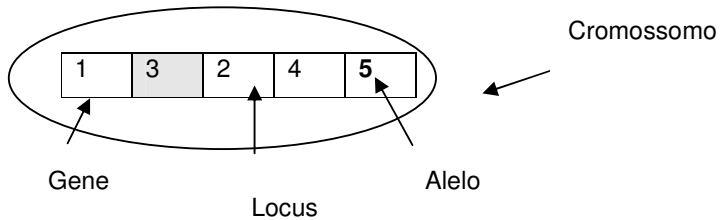


Figura 2 – Componentes da classe dos Algoritmos Evolutivos

Apesar de existirem várias técnicas de algoritmos evolutivos, pode-se definir um algoritmo básico para o mesmo, conforme a figura 3:

```

1 INICIALIZAR  $p(t)$  // população inicial;
2 AVALIAR  $p(t)$ ;
3  $t \leftarrow 0$  // número de gerações;
4 ENQUANTO critério_de_parada = falso FAÇA
5   ALTERAR( $p$ );
6   AVALIAR( $p$ );
7   SELECIONAR( $p$ );
8    $t \leftarrow t + 1$ ;
9 FIMDOENQUANTO

```

Figura 3 – Algoritmo Evolutivo Básico

Esse algoritmo tem no passo 1 a inicialização da população de soluções, executada nesse trabalho através de um algoritmo construtivo aleatorizado. No passo 2 será feita a avaliação dos indivíduos gerados anteriormente, verificados através da função objetivo com base no custo de cada rota no PCVDH. No passo 3 tem-se a inicialização do número de gerações, que poderá servir como critério de parada no passo 4. No passo 5 a população sofre uma alteração nos indivíduos da

população, através de recombinação ou mutação sofrida pelos mesmos, provocando modificações em seu estado com a geração de novos indivíduos. Esses novos indivíduos serão reavaliados no passo 6, e no passo 7 competirão com os restantes sua permanência na população. A iteração do passo 5 até o passo 8 acontece enquanto o critério de parada não seja satisfeito.

Várias abordagens para sistemas computacionais inspirados na evolução das espécies foram propostas, todas elas utilizando os princípios do algoritmo básico evolutivo da figura 2, e podemos classificá-las em quatro principais técnicas, conforme descrita a seguir.

3.1 Algoritmos Genéticos

Foram introduzidos por J. Holland (HOLLAND, 1975), utilizando os operadores de crossover (recombinação) e mutação para representar a reprodução.

A idéia básica era gerar indivíduos geneticamente superiores aos seus antecedentes, ou seja, a partir de soluções iniciais de um determinado problema, devem-se gerar soluções melhores do que as soluções antecedentes, considerando a função de aptidão (fitness), que nesse caso é a função objetivo.

3.2 Programação Evolutiva

Introduzida por Fogel (1966), foi inicialmente proposta como uma técnica para criar inteligência artificial pela evolução de máquinas de estado finito (empregando, também, apenas mutação).

Recentemente, tem sido aplicada a problemas de otimização, sendo, neste caso, virtualmente equivalente às estratégias evolutivas. Atualmente, existem apenas pequenas diferenças no que diz respeito aos procedimentos de seleção e codificação de indivíduos presentes nestas duas abordagens (FOGEL, 1994).

3.3 Programação Genética

Desenvolvida inicialmente em 1992 por J. Koza (KOZA, 1992), sendo aplicado como uma extensão dos algoritmos genéticos, herdando sua estrutura de dados, porém utilizando a estrutura de árvores, adaptando seus operadores de recombinação e mutação para as mesmas.

A diferença básica para os algoritmos genéticos é que na Programação Genética são usadas técnicas automáticas de programação que propiciam a evolução de programas de computadores para resolverem problemas.

3.4 Estratégias Evolutivas

As Estratégias Evolutivas tiveram origem na Universidade Técnica de Berlim em 1964, mas somente foram desenvolvidas em 1973 por Rechenberg (1973). Foram inicialmente propostas com o objetivo de solucionar problemas de otimização de parâmetros discretos e contínuos, empregando apenas os operadores de seleção e mutação, somente com um indivíduo da população. Em 1981 Schwefel introduziu o operador de recombinação, trabalhando com populações de mais de um indivíduo (SCHWEFEL, 1981).

As Estratégias Evolutivas serão o foco principal desse trabalho, as quais terão como implementação quatro de seus algoritmos. Por isso serão descritas nas seções 3.4.1 e 3.4.2 uma teoria sobre elas e um referencial dessas estratégias como metaheurísticas, respectivamente, bem como uma abordagem sobre as estratégias evolutivas ES1 e ES2, pois as mesmas serviram de inspiraram para a realização dessa dissertação.

3.4.1 Teoria das Estratégias Evolutivas

As Estratégias Evolutivas (EEs) foram desenvolvidas com o propósito inicial de resolverem os problemas de otimização de parâmetros contínuos na área da engenharia. A primeira EE desenvolvida foi a EE-(1+1), proposta por I. Rechenberg e H. P. Schwefel, nos anos 60 (RECHENBERG, 1965; SCHWEFEL, 1965), no Hermann Föttinger Institute for Hydrodynamics (Universidade Técnica de Berlim, Alemanha), em experimentos com um processo túnel de vento. A EE-(1+1) original utiliza somente o operador de mutação, onde apenas uma solução ancestral produz um único descendente por geração. A EE-(1+1) foi progressivamente generalizada em variantes do número de ancestrais (pais), $\mu > 1$, e número de descendentes (filhos), $\lambda > 1$, por geração. As EEs com múltiplos membros têm o embasamento biológico relacionado às características de poligenia e pleiotropia. Estas EEs são divididas de acordo com o mecanismo de seleção em:

- estratégia soma (plus strategy) ou EE-($\mu+\lambda$): Os μ ancestrais geram λ descendentes.

Após os μ ancestrais e os λ descendentes competem pela sobrevivência;

- estratégia vírgula (comma strategy) ou EE-(μ, λ): Os λ descendentes competem para sobreviver e os μ ancestrais são completamente substituídos a cada geração.

A EE-(μ, λ) tem a tendência de manter uma maior diversidade de indivíduos na população, o que pode ser uma vantagem de forma a evitar-se mínimos locais. A desvantagem da EE-(μ, λ) é a possibilidade de uma “solução ótima” obtida, em uma dada geração, não “sobreviver” até o final do procedimento evolutivo.

A EE-($\mu+\lambda$), ao contrário, da EE-(μ, λ), dependendo de sua configuração pode ser mais susceptível a mínimos locais, ou seja, o domínio de uma elite de ancestrais (soluções similares) antes da obtenção de um valor adequado. Entretanto, uma solução ótima obtida, durante o procedimento evolutivo, não é perdida e “sobrevive” até que o final do procedimento evolutivo (BÄCK et alii, apud COELHO, 2003).

3.4.2 Estratégias Evolutivas como Metaheurísticas

As metaheurísticas utilizadas nesse trabalho foram baseadas no trabalho de Jörg Homberger e Hermann Gehring (HOMBERGER; GEHRING, 1999), os quais desenvolveram as estratégias evolutivas (do inglês *Evolution Strategies*) ES1 e ES2. Essas metaheurísticas foram adaptadas para a resolução do Vehicle Routing Problem With Time Windows – VRPTW. Segundo Homberger e Gehring já foram obtidos bons resultados para o VRP (Vehicle Routing Problem) e para o VRPTW utilizando metaheurísticas como Busca Tabu (OSMAN, 1993; TAILLARD et al., 1996; CHIANG, RUSSELL, 1997; LIU, SHEN, 1998), *simulated annealing* (CHIANG; RUSSELL, 1996) e algoritmos genéticos (THANGIAH et al., 1991).

A solução do VRPTW por meio de estratégias evolutivas ainda não tem sido muito reportada na literatura, porém Ablay (1979) e Nissen (1994) as utilizaram para resolverem outros problemas de otimização combinatória. As estratégias evolutivas utilizadas por Homberger e Gehring serão descritas resumidamente a seguir nas seções 3.4.2.1 e 3.4.2.2. Para maiores detalhes sobre esses algoritmos evolutivos consultar (HOMBERGER; GEHRING, 1999).

3.4.2.1 Evolution Strategy ES1

Essa estratégia não utiliza recombinação nem código de mutação para representar a evolução. Conforme se pode ver na figura 4 um novo indivíduo é gerado de acordo com uma regra de mutação.

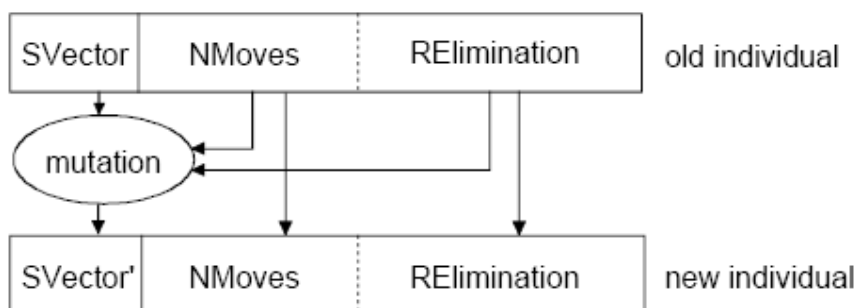


Figura 4 – Geração de um novo indivíduo através da *evolution strategy* ES1
 Fonte: (HOMBERGER; GEHRING, 1999)

O *SVector* da figura XX representa um vetor com uma possível solução para o VRPTW (uma rota). Essa rota, que foi escolhida aleatoriamente em *SVector*, recebe um dos operadores de troca: *Or-opt*, *2-opt** ou *1-interchange*, os quais fazem parte do *MoveSet* (conjunto de operadores de troca de rota). O parâmetro *NMoves* indica com que frequência o operador de trocas é utilizado na mutação, utilizando o intervalo $[1, \dots, 10]$. Os testes realizados por Homberger e Gehring demonstram que valores próximos ao limite superior apresentaram melhores resultados do que valores próximos ao limite inferior, pois valores baixos convergem prematuramente para um ótimo local. Após feita a seleção e aplicada a regra de mutação, é trocado o pior indivíduo pelo indivíduo novo (caso o novo seja melhor que o pior).

O parâmetro *RElimination* $[0,1]$ é utilizado para direcionar a busca na mutação, sendo que o valor 0 significa a minimização da distância percorrida dentro da mutação e o valor 1 significa a minimização do número de veículos. Tanto o *RElimination* quanto o *NMoves* são passados sem variação para o indivíduo novo. A figura 5 resume a regra aplicada na *evolution strategy* ES1.

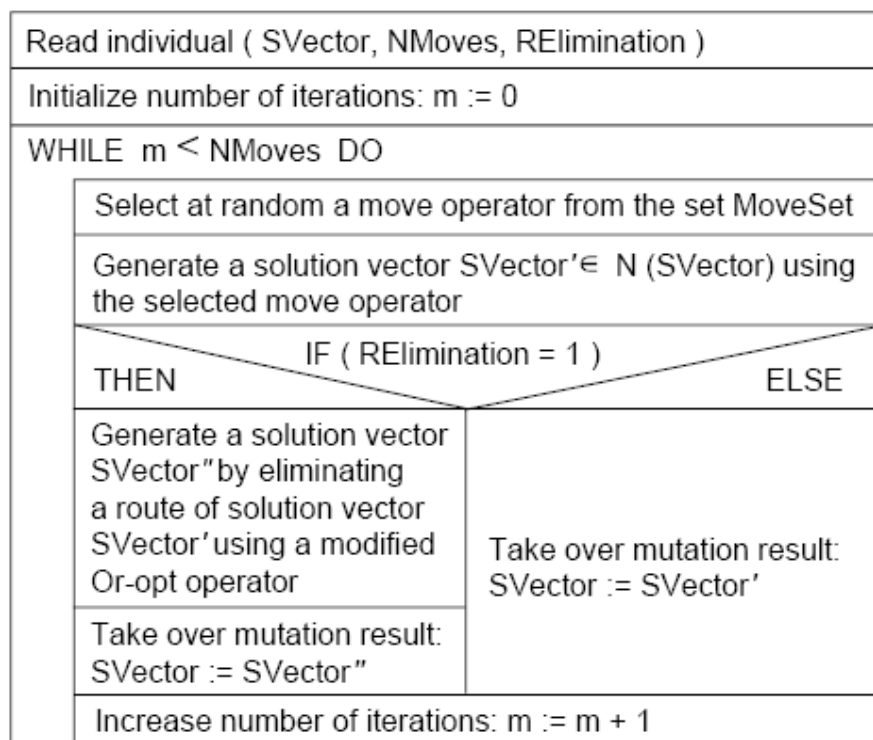


Figura 5 – Regra de mutação da *evolution strategy* ES1
 Fonte: (HOMBERGER; GEHRING, 1999)

3.4.2.2 Evolution Strategy ES2

A Metaheurística ES2 utiliza recombinação e mutação em seu algoritmo, procedimentos ausentes na ES1. Como mostra o exemplo da figura 6, partindo de três pais (SV1, SV2 e SV3) selecionados aleatoriamente, uma descendência é gerada em dois momentos consecutivos: em um 1º momento é gerada uma descendência temporária por recombinação do código de mutação de dois pais, resultando em um terceiro código de mutação denominado MC'3. A recombinação dos dois códigos de mutação é realizada utilizando o operador *uniform order-based crossover* (DAVIS, 1991). Após é feita a cópia do vetor de solução SV3 e do parâmetro RE₃ para a descendência temporária gerada. Esse processo de geração do código de mutação imita o princípio de reprodução sexuada, enquanto que o procedimento de cópia tem como base o princípio de replicação biológica (clonagem).

Em um 2º momento a descendência temporária é transferida para uma descendência definitiva, usando o terceiro pai (SV₃) para a mutação. A mutação é

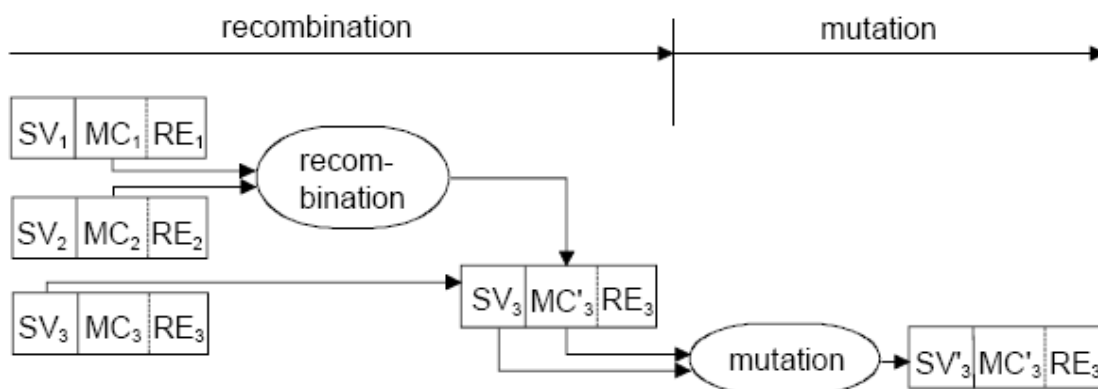


Figura 6– Exemplo de geração de um novo indivíduo através da *evolution strategy* ES2
 Fonte: (HOMBERGER; GEHRING, 1999)

realizada através da aplicação do código de mutação, que consiste em um vetor de tamanho $2n$, onde n representa o número de vértice, isto é, cada vértice aparece duas vezes dentro do vetor. A primeira ocorrência do vértice indica uma operação de remoção e a segunda aparição identifica uma operação de inserção, conforme ilustra a figura 7.

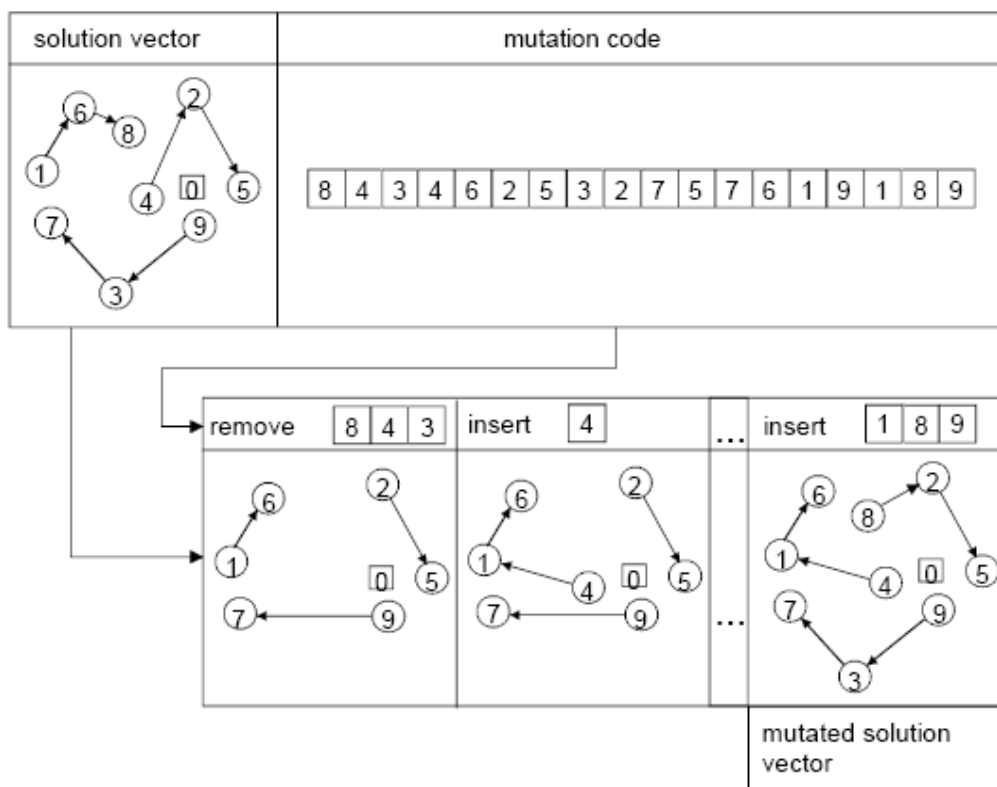


Figura 7 – Exemplo da mutação em um indivíduo
 Fonte: (HOMBERGER; GEHRING, 1999)

Este capítulo apresentou uma breve conceituação referente aos algoritmos pertencentes à computação evolutiva. Na próxima seção será apresentada a metodologia utilizada na resolução do PCVDH.

4 METODOLOGIA PARA A RESOLUÇÃO DO PROBLEMA

Esse capítulo apresenta a metodologia utilizada para a resolução do problema dessa dissertação, fazendo inicialmente uma explicação mais detalhada sobre as características do PCVDH, após é realizada uma abordagem dos seguintes algoritmos: construtivo, melhoramento e metaheurísticas (Estratégias Evolutivas).

4.1 Características do PCVDH

Considerando a formulação matemática apresentada na seção 2.5 para o PCVDH, deve-se, para um melhor entendimento de sua resolução, apresentar as peculiaridades desse problema.

O PCVDH possui Instâncias (ocorrências) do problema, que possuem o formato descrito na figura 8 , bem como sua respectiva explicação:

Nome do arquivo: **15_4_5_25.txt**

```

15 4           onde: 15 = número de cidades e 4 = cidade de origem

118 110 95 0 84 106 29 115 98 33 120 110 123 0 47 = Demandas

0 0 1177 0 0 2229 0 0 0 0 0 0 0 0 0
0 0 0 2023 0 0 0 0 2382 0 0 0 0 0 0
3085 0 0 545 0 1314 552 0 0 3099 0 1347 0 1221 0
0 1676 1668 0 0 1210 854 0 0 1762 0 0 1378 0 0
0 0 0 0 0 0 3171 919 0 0 0 0 0 2238
2180 0 2630 2848 0 0 0 1263 0 2200 0 0 0 0 0
0 0 2160 3051 0 0 0 2356 0 0 2855 0 1599 0 0
0 0 0 0 2477 2186 2313 0 0 2418 0 0 0 7 0
0 2223 0 0 3008 0 0 0 0 0 0 0 2057 0 0
0 0 2388 3082 0 2508 0 2458 0 0 0 0 0 0 0
0 0 0 0 0 1902 0 0 0 0 1169 0 1282 0
0 0 2487 0 0 0 0 0 0 1926 0 0 0 0
0 0 0 2857 0 0 1925 0 2458 0 0 0 0 0 0
0 0 2740 0 0 0 0 2076 0 0 2168 0 0 0 2123
0 0 0 0 2533 0 0 0 0 0 0 0 1781 0

0 0 4 0 0 4 0 0 0 0 0 0 0 0 0
0 0 0 8 0 0 0 0 7 0 0 0 0 0 0
6 0 0 1 0 2 2 0 0 3 0 5 0 3 0
0 3 6 0 0 1 1 0 0 1 0 0 1 0 0
0 0 0 0 0 0 0 6 1 0 0 0 0 0 4
6 0 7 1 0 0 0 2 0 1 0 0 0 0 0
0 0 1 6 0 0 0 9 0 0 5 0 6 0 0
0 0 0 0 1 2 1 0 0 9 0 0 0 1 0
0 6 0 0 9 0 0 0 0 0 0 0 6 0 0
0 0 2 3 0 2 0 1 0 0 0 0 0 0 0
0 0 0 0 0 0 5 0 0 0 0 2 0 1 0
0 0 7 0 0 0 0 0 0 3 0 0 0 0 0
0 0 0 11 0 0 7 0 1 0 0 0 0 0 0
0 0 1 0 0 0 0 1 0 0 8 0 0 0 6
0 0 0 0 7 0 0 0 0 0 0 0 7 0

```

Matriz de Custo Fixo,
representando:
↓
nº de cidades x nº de cidades
Nesse caso é uma matriz 15x15

Matriz de Custo Variável,
representando:
nº de cidades -1 matrizes

Figura 8 – Exemplo de uma instância do PCVDH

A primeira linha da instância possui dois números, sendo que o primeiro representa o número de cidades da matriz e o segundo a cidade de origem, pois esse problema sempre tem uma determinada cidade de partida já definida, não é escolhida aleatoriamente.

O próximo elemento é uma matriz representando os custos fixos das cidades. Nesse problema de entrega de mercadorias, o custo fixo representa o custo que um caminhão tem para percorrer uma determinada estrada, mesmo que este caminhão não tenha nenhum produto. Por fim as matrizes de custo variável, que contêm o *número de cidades – 1 matrizes*, pois a cidade de origem não possui matriz. Essa

matriz contém o custo variável para transportar um produto k da cidade i até a cidade j . No exemplo da figura 8 ter-se-ia custos para as cidades

$1 - 2 - 3 - 5 - 6 - 7 - 8 - 9 - 10 - 11 - 12 - 13 - 14 - 15$

pois a cidade 4 é a cidade de origem.

Cada instância do problema é representada por um arquivo (na figura XX é 15_4_5_25.txt) que possui o formato padrão $A_B_C_D$, com a seguinte convenção: A representa o número de vértices, B o vértice de origem, C é um percentual significando que o custo variável é no máximo $C\%$ do custo fixo, e D indica o percentual de conectividade do grafo.

Diz-se que um grafo é totalmente conexo (figura 9a) se existe uma aresta E em qualquer par de vértices V pertencente ao grafo G . Se não existir pelo menos uma aresta entre qualquer par de vértices, então este grafo é denominado desconexo (figura 9b).

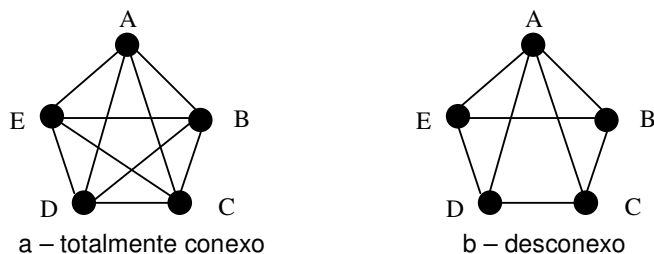


Figura 9 – Exemplo de conectividade dos grafos

As instâncias do PCVDH são divididas em baixa densidade e alta densidade. As instâncias de baixa densidade caracterizam-se por ter um percentual de conectividade de até no máximo 25%. Por outro lado, as instâncias de alta densidade possuem 70% de conectividade.

Cabe salientar que o método exato utilizado por Sarubbi não conseguiu encontrar nenhum resultado com as instâncias de alta densidade. Isso se explica porque, devido ao alto grau de conectividade e tendo esse algoritmo o tempo como limitante, o mesmo não consegue enumerar todas as possíveis rotas.

Quando uma instância não é totalmente conectada (figura 10a), determinadas rotas não são possíveis de serem ligadas devido à ausência de arestas entre determinados pares de vértices. Uma solução para esse problema é a inserção de arestas que faltam para tornar uma rota 100% conectada. Essas arestas são

chamadas de arestas fantasmas ou arestas inactiváveis, ou ainda arcos fantasmas (figura 10b), pois violam as restrições do problema.

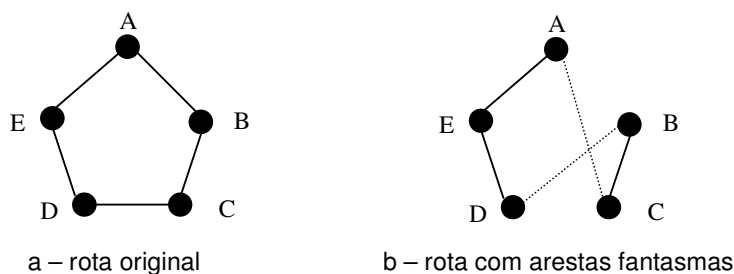


Figura 10 – Exemplo de grafo com arestas fantasmas

O intuito dessa medida é fazer com que o algoritmo seja capaz de partir de soluções iniciais inactiváveis e obter soluções factíveis. Para isso é atribuído a um arco fantasma um custo muito grande, nesse caso tendendo ao infinito, fazendo com que todas as outras possíveis soluções sejam testadas. Assim, quando essa rota fantasma for testada, provavelmente a solução inicial do problema já estará formada.

Os algoritmos utilizados para a proposta de resolução do problema foram divididos em três etapas:

- na **1ª etapa** foi definida uma estrutura de lista circular duplamente encadeada, sendo utilizado o algoritmo de inserção aleatorizado do Vizinho Mais Próximo na construção da solução inicial (população) para a resolução do PCVDH;
- na **2ª etapa** foram desenvolvidos algoritmos de melhoramento ou aprimoramento, que são aplicados a partir de soluções iniciais obtidas pelas heurísticas construtivas, utilizando técnicas k-opt, mais precisamente algoritmos 2-opt e 3opt.
- a **3ª etapa** será percorrida com uma metaheurística baseada em algoritmos evolutivos, a qual atuará como uma busca local explorando o espaço de soluções, para além do ótimo local, buscando boas soluções e novas regiões promissoras.

4.2 Heurística do Vizinho Mais Próximo

A Heurística do Vizinho Mais Próximo – VMP, é uma heurística simples onde uma determinada rota é inicializada por um vértice (cidade inicial), após será

procurado o vértice mais próximo do último vértice adicionado e será criado um caminho (aresta) entre esses dois vértices. Quando todos os vértices estiverem no caminho, é inserida uma aresta conectando o último vértice adicionado ao vértice inicial.

A matriz de cidades (5x5) da figura 11 contém as distâncias simétricas entre cada par de cidades para demonstração do algoritmo construtivo.

Cidades	1	2	3	4	5
1	0	1	2	7	5
2	1	0	3	4	3
3	2	3	0	5	2
4	7	4	5	0	5
5	5	3	2	3	0

Figura 11 - Matriz de cidades x cidades

Os movimentos utilizados para a construção de uma rota inicial (figura 12) estão descritos abaixo:

- i. escolha do vértice inicial: **1**
- ii. 2º vértice a ser inserido na rota: **2**
- iii. 3º vértice a ser inserido na rota: **5**
- iv. 4º vértice a ser inserido na rota: **3**
- v. 5º vértice a ser inserido na rota: **4**
- vi. retorna do último vértice ao vértice inicial

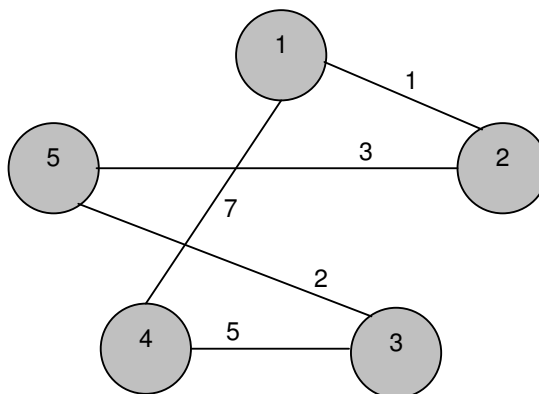


Figura 12 – Geração de uma rota inicial

4.3 Heurística de Melhoria *k-Optimal* (*k-opt*)

Após a realização da 1ª etapa, através do algoritmo construtivo aleatorizado, tem-se um ciclo Hamiltoniano com a seqüência dos vértices visitados. A idéia agora passa a ser de se conseguir uma melhora no valor da função de avaliação através da troca na ordem de visita dos vértices, ou seja, mudando-se a conexão das arestas de *k*-vértices. Essa heurística é derivada dos trabalhos de Lin e Kernighan (LIN, 1965 e LIN;KERNIGHAN, 1973).

A seguir se tem uma descrição dos três algoritmos de melhoria utilizados nessa dissertação.

4.3.1 Heurística 2-opt

Este algoritmo testa a remoção de dois arcos não adjacentes e a ligação dos nós restantes de forma diferente da original, conforme os passos de remoção e inserção de arcos na figura 13. Basicamente, esse algoritmo é composto dos seguintes passos nesse trabalho:

1. Considerar uma rota inicial (solução inicial) contendo as matrizes de custo fixo e custo variável;
2. Selecionar duas arestas não adjacentes na rota da solução inicial;
3. Verificar, através de movimentos de troca de arestas, se o custo após as trocas é menor que o custo inicial (o custo nesse caso é a função objetivo);
4. Repetir o passo 2 até que nenhum movimento aprimorante seja possível.

Apresentando de uma forma mais detalhada o algoritmo citado anteriormente tem-se os seguintes passos:

* Considerando que um arco tem os nós anterior e posterior.

Passo 1: Faz *i* e *j* serem os dois primeiros arcos não adjacentes, respectivamente.

Passo 2: Remove *i* e *j*.

Passo 3: Liga *i*.anterior com *j*.anterior.

Passo 4: Troca a direção do trecho *i*.posterior até *j*.anterior.

Passo 5: Liga *i*.posterior com *j*.posterior.

Passo 6: Se o custo da rota resultante for menor que o custo da MELHOR_ROTA, faz MELHOR_ROTA = rota.

Passo 7: Se *j* não for anterior a *i*, "anda" com *j* e volta ao **Passo 2**.

Passo 8: Se i não for o último arco da rota, "anda" com i , faz j ser o arco posterior mas não adjacente a i e volta ao **Passo 2**.

Passo 9: A rota resultante é MELHOR_ROTAS.

Após a execução de todos os movimentos de troca de arestas se chega ao ótimo local. Com a realização de todas essas trocas obtém-se a exploração de soluções vizinhas dentro do espaço de busca.

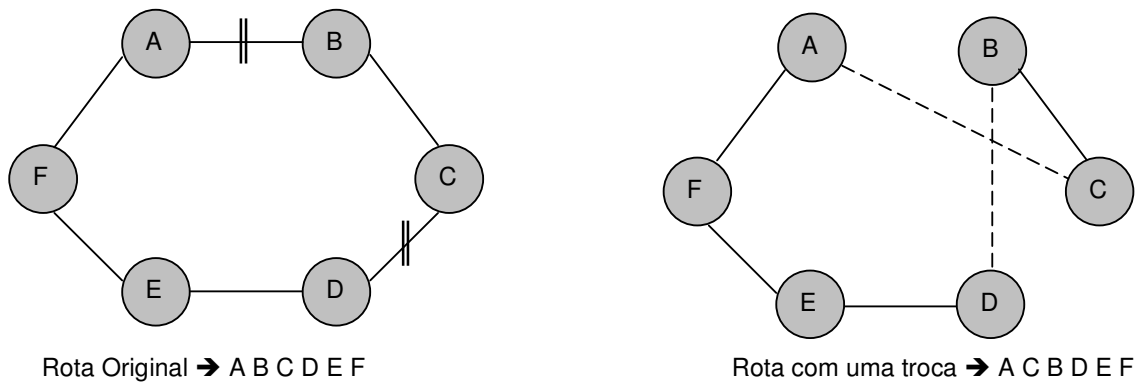
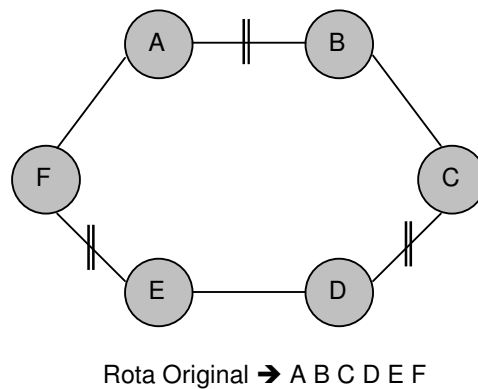
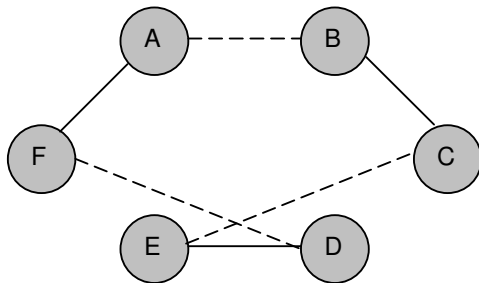


Figura 13 – Um movimento de troca 2-opt válido

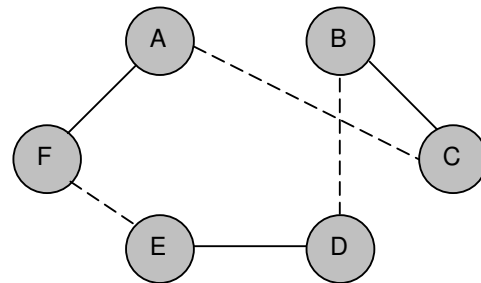
4.3.2 Heurística 3-opt

A heurística 3-opt é semelhante à heurística 2-opt, com a diferença de que a primeira testa a troca de três arcos distintos. Os grafos da figura 14, representados a seguir, demonstram o funcionamento do movimento 3-opt:

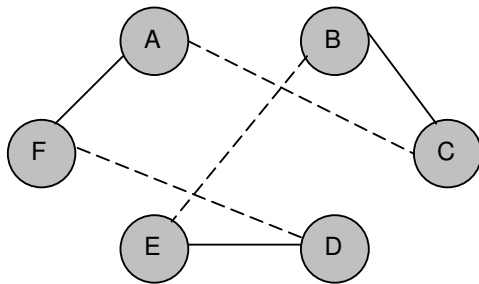




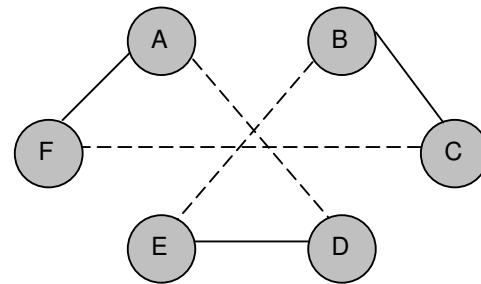
1ª Troca → A B C E D F



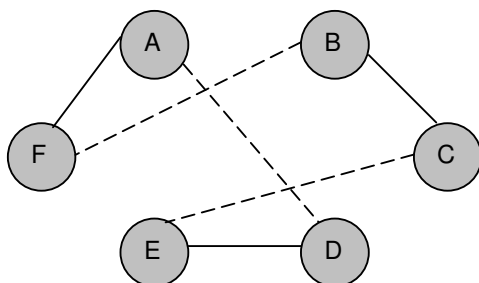
2ª Troca → A C B D E F



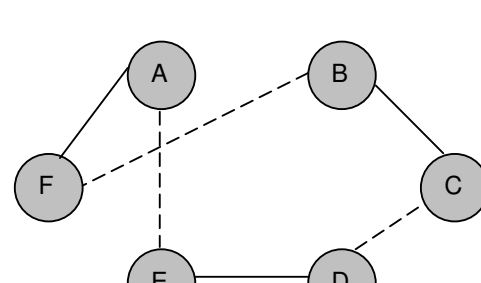
3ª Troca → A C B E D F



4ª Troca → A D E B C F



5ª Troca → A D E C B F



6ª Troca → A E D C B F

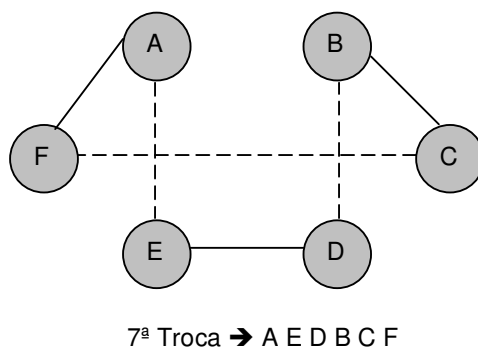
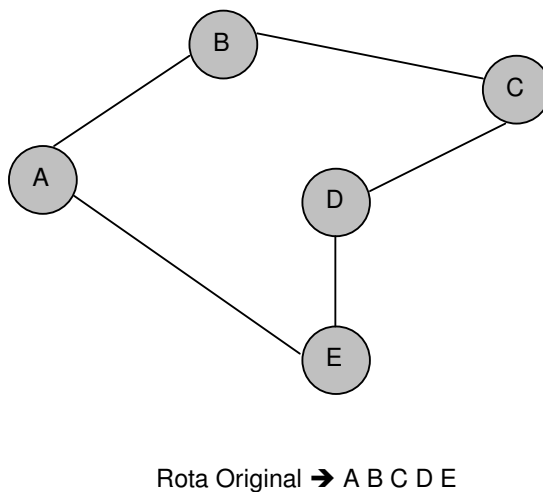
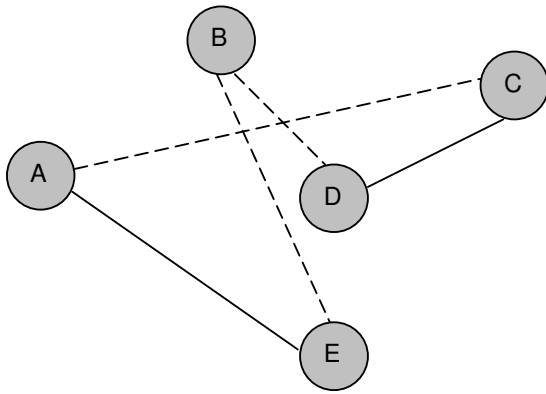


Figura 14 - Movimentos de trocas de arcos no algoritmo 3-opt

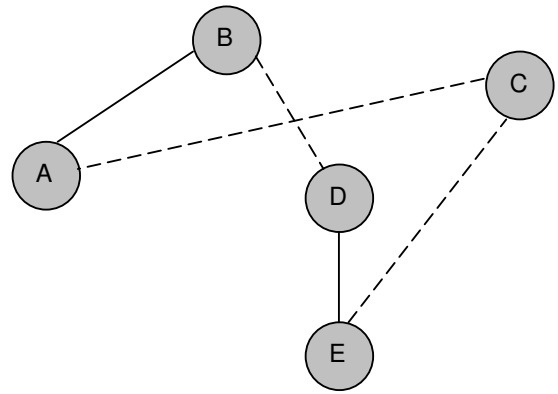
4.3.3 Heurística Or-opt

A idéia básica da troca or-opt é reinserir segmentos de vértices consecutivos em outros locais dentro da rota e aceitar o movimento se a inserção melhorar o valor da solução corrente. Repete-se o procedimento de eliminação até que nenhuma rota mais possa eliminada. Também nesse algoritmo foi utilizado um construtivo aleatorizado para gerar uma solução inicial. A figura 15 demonstra os movimentos desse algoritmo. Mais detalhes podem ser encontrados em (POTVIN; KERVAHUT; GARCIA; ROUSSEAU, 1996).

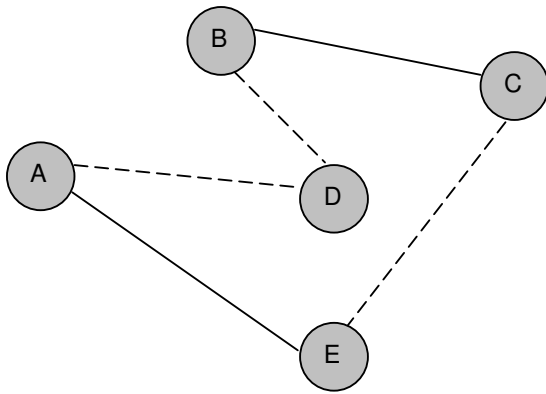




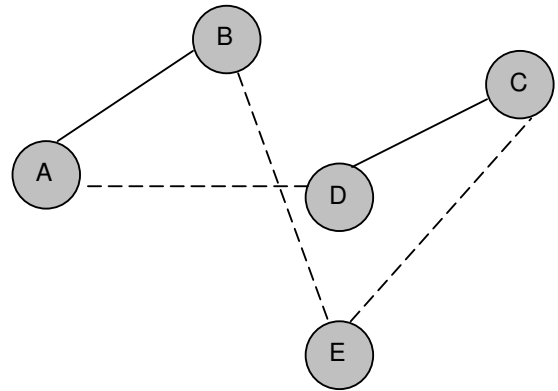
1ª Troca → A, B, C



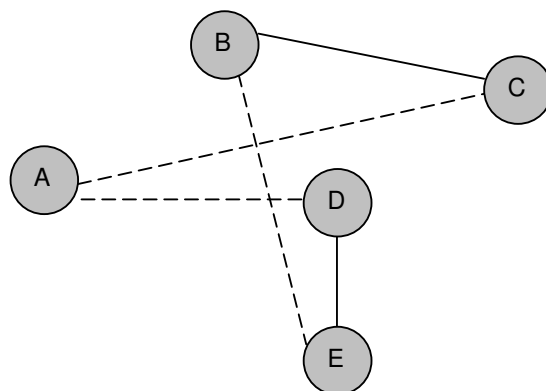
2ª Troca → B, C, D



3ª Troca → C, D, E



4ª Troca → D, E, A



5ª Troca → E, A, B

Figura 15 – Movimentos de troca de um algoritmo or-opt

4.4 Teoria das Estratégias Evolutivas

As Estratégias Evolutivas (EEs) foram desenvolvidas com o propósito inicial de resolverem os problemas de otimização de parâmetros contínuos na área da engenharia. A primeira EE desenvolvida foi a EE-(1+1), proposta por I. Rechenberg e H. P. Schwefel, nos anos 60 (RECHENBERG, 1965; SCHWEFEL, 1965), no Hermann Föttinger Institute for Hydrodynamics (Universidade Técnica de Berlin, Alemanha), em experimentos com um processo túnel de vento. A EE-(1+1) original utiliza somente o operador de mutação, onde apenas uma solução ancestral produz um único descendente por geração.

A EE-(1+1) foi progressivamente generalizada em variantes do número de ancestrais (pais), $\mu > 1$, e número de descendentes (filhos), $\lambda > 1$, por geração. As EEs com múltiplos membros têm o embasamento biológico relacionado às características de poligenia e pleiotropia. Estas EEs são divididas de acordo com o mecanismo de seleção em:

- estratégia soma (plus strategy) ou EE-($\mu + \lambda$): Os μ ancestrais geram λ descendentes.

Após os μ ancestrais e os λ descendentes competem pela sobrevivência;

- estratégia vírgula (comma strategy) ou EE-(μ, λ): Os λ descendentes competem para sobreviver e os μ ancestrais são completamente substituídos a cada geração.

A EE-(μ, λ) tem a tendência de manter uma maior diversidade de indivíduos na população, o que pode ser uma vantagem de forma a evitar-se mínimos locais. A

desvantagem da $EE-(\mu, \lambda)$ é a possibilidade de uma “solução ótima” obtida, em uma dada geração, não “sobreviver” até o final do procedimento evolutivo.

A $EE-(\mu+\lambda)$, ao contrário, da $EE-(\mu, \lambda)$, dependendo de sua configuração pode ser mais susceptível a mínimos locais, ou seja, o domínio de uma elite de ancestrais (soluções similares) antes da obtenção de um valor adequado. Entretanto, uma solução ótima obtida, durante o procedimento evolutivo, não é perdida e “sobrevive” até que o final do procedimento evolutivo (BÄCK et alii, apud COELHO, 2003).

4.4.1 Estratégia Evolutiva ES1

A estratégia evolutiva ES1 (figura 16) é uma metaheurística que, similarmente aos algoritmos genéticos, manipula uma população de indivíduos, os quais representam soluções para o problema de otimização combinatória, nesse trabalho o PCVDH.

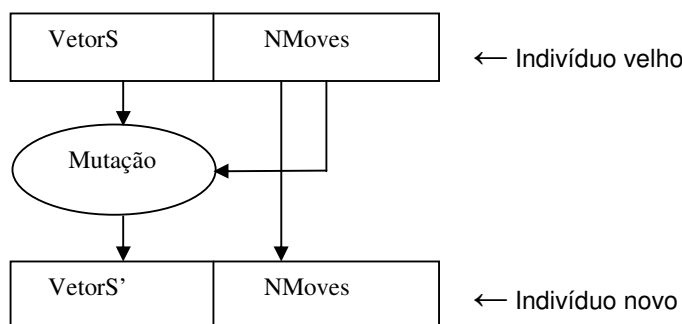


Figura 16 – Geração de um novo indivíduo utilizando a Estratégia Evolutiva ES1
Adaptado de HOMBERGER; GEHRING (1999)

A seguir, na figura 17, são apresentados os passos para a execução do algoritmo ES1:

- PASSO 1** - Receber parâmetros (VS, NMoves, TxC);
- PASSO 2** - Inicializar o Contador de **Iterações**;
- PASSO 3** - Enquanto o **Iterações** < **NMaxlte** (quantidade de melhoramentos) Faça
 Selecionar randomicamente um operador do MoveSet;
 Gerar um VS' usando o operador selecionado no item anterior com o
 parâmetro NMoves;
 Inserir VS' na população;
 Se **Iterações** >= **icresc**
 Remover o pior indivíduo da população;
 Incrementar Iterações;
- PASSO 4** - Parar a execução, é a melhor solução para o ES1..

Figura 17 – Passos da execução do algoritmo ES1

O funcionamento dessa estratégia dá-se da seguinte forma: no passo 1 são recebidos os parâmetros de controle do algoritmo, que são o Vetor Solução (VS) – que contém a solução inicial do problema, a qual foi criada utilizando o algoritmo construtivo aleatorizado VMP, visto anteriormente, o NMoveS – que indica qual a frequência máxima de melhoramentos a cada execução do operador escolhido no MoveSet, e o Índice de Crescimento (icresc), indicando em quanto a população pode sofrer crescimento; no passo 2 é inicializado um contador de iterações. No passo 3 é executado um laço enquanto as iterações forem menores que a quantidade de melhoramentos, onde seleciona-se randomicamente um operador do MoveSet (conjunto de algoritmos de melhoramento), gera-se um novo indivíduo através do operador de mutação (Or-opt, 2 opt ou 3-opt), o qual é inserido na população. Depois de feita a seleção e a mutação, se as iterações forem maiores ou igual ao Índice de Crescimento, é removido o pior indivíduo da população. No passo 4 pára a execução, pois essa é a melhor solução para o ES1.

O parâmetro NMoves fixa o tamanho da mutação, não tendo influência na sucessão da mutação. O NMoves é passado sem variação para o indivíduo novo.

4.4.2 Estratégia Evolutiva ES2

A estratégia Evolutiva ES2 é executada fazendo uma conexão entre recombinação e mutação, características ausentes na ES1. Partindo de três pais uma descendência temporária é gerada em dois passos consecutivos (ver figura 18):

- no 1º passo é realizada a recombinação dos códigos de mutação (cada indivíduo da população possui um código de mutação) de dois pais (CM_1 e CM_2), para gerar um filho com as características dos pais, resultando em um terceiro código de mutação (CM'_3); é realizada também a cópia do vetor solução do terceiro pai (VS_3) para a descendência gerada;

- no 2º passo a descendência temporária é transferida para uma descendência definitiva, utilizando o terceiro pai (VS_3) para a mutação (CM'_3).

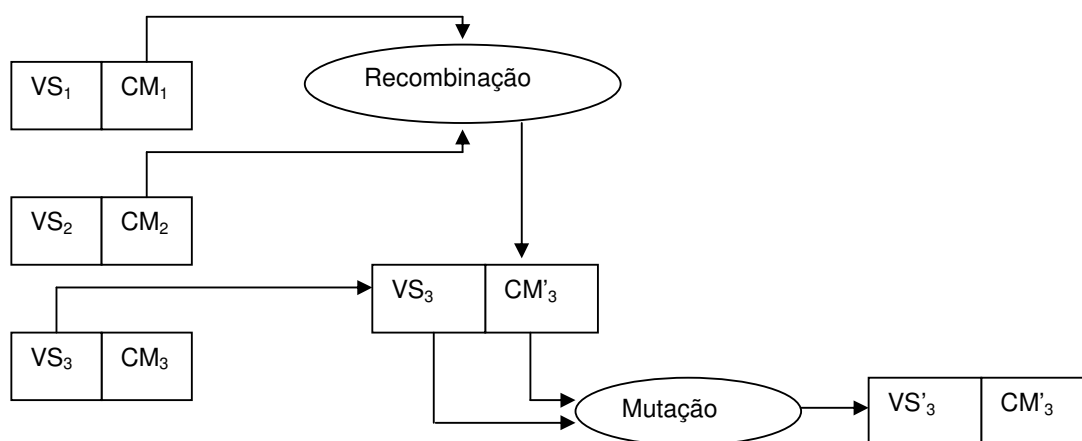


Figura 18 – Geração de um novo indivíduo utilizando a Estratégia Evolutiva ES2
Adaptado de (HOMBERGER; GEHRING, 1999)

O código de mutação do vetor é uma lista de vértices de tamanho $2n$, sendo que cada vértice aparece duas vezes na lista. Primeiramente é realizada uma cópia do vetor solução. Esse vetor é percorrido da esquerda para a direita. A cada posição do vetor é verificado se o número da cidade ocorre pela primeira ou segunda vez. Se for a primeira vez, uma operação de remoção é executada, caso contrário uma operação de inserção é executada. O algoritmo de inserção utilizado é o da Inserção Mais Barata (GOLDBARG; LUNA, 2000).

O operador de *crossover* (recombinação) utilizado é o *Position Based Crossover* (PBX) (STARKWEATHER, 1991), conforme a ilustração da figura 19. Esse algoritmo recebe como entrada dois vetores representando os códigos de

mutação dos pais. O PBX começa selecionando um número de posições aleatórias, sendo que ele impõe, nas posições selecionadas, que o filho tenha os mesmos elementos do pai2. Os elementos restantes do filho vêm do pai1, mantendo a mesma ordem presente no pai1.

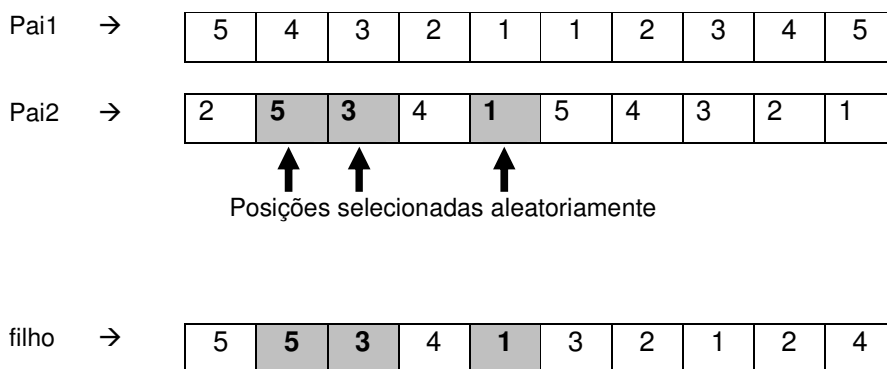


Figura 19 – Execução do operador de recombinação PBX

Os passos para a execução do ES2 são os seguintes:

- PASSO 1** - Receber parâmetros (VS, TxC)
- PASSO 2** - Inicializar o Contador de **Iterações**
- PASSO 3** - Gerar o código de mutação para os indivíduos da população
- PASSO 4** - Se **Iterações** > **NMaxIte Pare**. Caso contrário selecione randomicamente dois indivíduos (pais): P1 e P2, considerando M1 e M2 o código de mutação de P1 e P2, respectivamente
- PASSO 5** - Gerar um novo código de mutação denominado M3', proveniente da recombinação de M1 e M2
- PASSO 6** - Buscar aleatoriamente na população o indivíduo P3
- PASSO 7** - Realizar a mutação em P3 através do código de mutação M3', resultando em um novo indivíduo P3'
- PASSO 8** - Inserir P3' na população, associando M3' a P3'.
- PASSO 9** - Se **Iterações** >= **TxC**

Remover o pior indivíduo da população
- PASSO 10** - Incrementar Iterações e ir para o **PASSO 4**.

Figura 20 – Passos de execução do algoritmo ES2

4.4.3 Estratégia Evolutiva ES3

Essa nova estratégia proposta é semelhante a estratégia ES1, isto é, os passos são os mesmos executados no algoritmo ES1 (figura 10), sendo que a mudança decorre em função da retirada do algoritmo de melhoramento *or-opt* e a inclusão do algoritmo *swap* dentro do *MoveSet*.

A troca utilizada é a *2-swap* (figura 21), onde são feitas todas as combinações possíveis envolvendo trocas entre dois vértices.

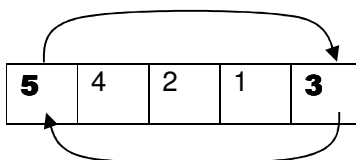


Figura 21 – Demonstração de uma troca *swap*

Esse algoritmo foi criado com a finalidade de avaliar os resultados dos custos após cada iteração trabalhando com a hipótese de que, depois de efetuada uma troca entre cidades, pode-se ter escolhido aleatoriamente uma cidade mais distante na rota, tendo essa cidade uma penalidade maior. Essa cidade sendo visitada no início do caminho faz com que o caminhão percorra o restante do percurso com uma carga menor, conseqüentemente minimizando o custo total da rota.

4.4.4 Estratégia Evolutiva ES4

A Estratégia Evolutiva ES4 é uma hibridização da ES1 (seção 4.3.1) com a ES2 (seção 4.3.2). A estratégia ES4 tem sua execução idêntica à ES2 até o passo 7, como é descrito a seguir:

- inicialmente é gerado o código de mutação para os indivíduos da população. Se o número de iterações for maior que o número máximo de iterações, pára o algoritmo, caso contrário, são selecionados randomicamente dois indivíduos (pais): P1 e P2, considerando M1 e M2 o código de mutação de P1 e P2, respectivamente. Gera-se um novo código de mutação denominado M3', proveniente da recombinação de M1 e M2. É buscado aleatoriamente na população um terceiro indivíduo – P3. Realiza-se então a mutação em P3 através do código de mutação M3', resultando em novo indivíduo P3'.

A partir desse momento entra em funcionamento uma parcela do algoritmo ES1 (Passo 3, seção 4.3.1), que é executado em cima do indivíduo P3':

- enquanto o número máximo de aprimoramentos for menor que o número de aprimoramentos (NMoves) é selecionado aleatoriamente um operador do conjunto de operadores (MoveSet). É gerado um novo vetor solução, o qual é inserido na população. Se o número de iterações for maior ou igual ao tamanho máximo da população é removido o pior indivíduo da população e incrementa-se o número de iterações.

Este capítulo apresentou as heurísticas utilizadas nessa dissertação, fazendo um estudo mais detalhado sobre as estratégias evolutivas propostas para a resolução do problema. O capítulo seguinte irá apresentar os resultados computacionais e análise dos resultados obtidos.

5 RESULTADOS COMPUTACIONAIS

Este capítulo apresenta os resultados computacionais obtidos pelos algoritmos evolutivos abordados nas seções anteriores. Os testes desses algoritmos foram realizados em um microcomputador AMD Sempron™ 3000+, 2.0 GHz, com 512 Mb de memória RAM, rodando em uma máquina com sistema operacional Microsoft Windows XP. A linguagem utilizada foi Java, da Sun Microsystem, versão J2SDK 5.0, rodando sobre máquina virtual Java.

As tabelas de custos que contêm os testes computacionais possuem colunas, onde cada item do cabeçalho será explicado a seguir:

- Instância – conforme definido em Sarubbi (2003) o padrão possui o formato **A_B_C_D**, sendo que A o número de vértices, B o vértice de origem, C é um percentual significando que o custo variável é no máximo C% do custo fixo, e D indica o percentual de conectividade do grafo;
- Sarubbi – apresenta os resultados dos testes exatos executados por Sarubbi, contendo as colunas **Lim.Inf.** (limite inferior) e **Lim.Sup.** (limite superior). Quando somente a coluna Lim.Inf. apresentar valor, então o algoritmo de Sarubbi encontrou a solução ótima, e mostra o custo na respectiva coluna. Se somente a coluna Lim.Sup. apresentar valor, é porque o algoritmo não encontrou a solução ótima para a instância. Constando valores nos dois limitantes, o algoritmo exato não chegou até o fim, portanto não encontrou a solução ótima, sendo que essa solução ficou entre esses dois valores. Por fim, quando não há valor em nenhuma das colunas o algoritmo não encontrou nenhuma solução.
- Algoritmos – apresenta a comparação dos custos e dos tempos para cada algoritmo nomeado no cabeçalho. Quando for o custo será mostrado o valor da função objetivo encontrado, se houver factibilidade, caso contrário será mostrado o número de arestas inactíveis sucedidos por *i*. Quando aparecer o custo com um número entre parênteses ao lado, significa que o algoritmo, após dez rodadas,

encontrou somente um resultado factível, exatamente na rodada representada pelo número entre parênteses. Existe também uma coluna que representa quão próxima a solução do algoritmo evolutivo está do algoritmo exato, representada em porcentagem (%).

O rodapé da tabela apresenta uma linha com os totais, onde o primeiro valor indica o número de arcos inactíveis e o segundo o número de instâncias que apresentaram inactibilidade, isto para cada parâmetro testado. Uma outra linha indica as médias, sendo que o primeiro valor é referente à média de arcos inactíveis, obtida através da divisão de arcos inactíveis pelo número de instâncias inactíveis, e o segundo valor refere-se à média da proximidade que o resultado alcançou referente à solução ótima, também para cada parâmetro testado.

Os tempos computacionais demandados pelos algoritmos estão descritos em segundos. Ainda nas tabelas relativas ao tempo de execução dos algoritmos evolutivos, contêm na ultima linha a média dos tempos de cada algoritmo para todos os parâmetros testados.

Por último salienta-se que os custos foram obtidos através da média de dez rodadas de cada instância porque os algoritmos tratam muito com aleatoriedade e diversidade da população, ocorrendo variação dos valores após cada iteração.

5.1 Algoritmo ES1

A estratégia evolutiva ES1 possui três parâmetros de entrada para a execução dos testes computacionais: o tamanho da população, a taxa de crescimento (opcional) e o número de gerações (iterações). Os parâmetros testados nesse trabalho foram: [10 50], [20 10 100] e [50 30 150], porque esses podem ser comparados com outro trabalho que aborda o mesmo problema. Também foram aplicadas outras configurações, mas que não produziram resultados satisfatórios ou então ficaram semelhantes às citadas anteriormente.

Para a execução dos testes computacionais de baixa densidade foram analisadas as 64 instâncias do trabalho de Sarubbi.

A execução das instâncias com os parâmetros [10 50] (tabela 1), isto é, sem crescimento da população, apresentou factibilidade em 87,50% dos casos, encontrando inactibilidade em apenas oito instâncias, e ficando com uma média de 1,932% em relação à proximidade dos resultados do algoritmo exato.

Tabela 1
Custos do ES1 para instâncias de 15 a 65 cidades [10 50]

Instância	Sarubbi, 2003		Pereira, 2004		ES1	
	Lim.Inf.	Lim.Sup.	10-50	%	10-50	%
15_4_25_25	128127		128127	0,000	128127	0,000
15_4_5_25	61515		61656	0,229	61515	0,000
15_4_5_25_1p	41481		41481	0,000	41481	0,000
15_7_1_15	33016		33016	0,000	33016	0,000
15_7_2_15	33892		33944	0,153	33899	0,021
18_7_1_17	37623		38748	2,990	37623	0,000
18_7_2_17	51869		51869	0,000	51869	0,000
18_7_3_15			2 i		(55367) (7)	
18_7_3_17	47877		47877	0,000	47877	0,000
20_18_1_13			1 i		1 i	
20_18_1_15	47450		47450	0,000	47450	0,000
20_18_3_15	53627	53630	53630	0,006	53627	0,000
20_18_3_15_1p	43970		44911	2,140	44021	0,116
20_18_5_15	67271		67271	0,000	67271	0,000
20_18_5_15_2p	51521		51521	0,000	51521	0,000
22_10_1_13	53233		53233	0,000	53233	0,000
22_10_2_13	47330		47330	0,000	47330	0,000
22_10_2_13_1p	45250		52639	16,329	49871	10,212
25_17_1_11	53134		53134	0,000	53134	0,000
25_17_2_11	64262		64536	0,4264	64262	0,000
25_17_3_11	76400		79170	3,6257	77001	0,787
25_17_3_11_1p	63426		65274	2,9136	63426	0,000
26_19_1_10	53358		53358	0,000	53358	0,000
26_19_1_13	54310	62616	59673	9,875	56793	4,572
28_20_1_10	71066		71066	0,000	71066	0,000
28_20_2_10	78054		78054	0,000	78054	0,000
28_20_2_10_1p	71964		1 i		75409	4,787
29_2_0_9	51128		51128	0,000	51128	0,000
29_2_1_9	79523		80979	1,831	79523	0,000
30_4_1_8	71198		71198	0,000	71198	0,000
30_4_1_8_1p			71198		70258	
30_4_2_8	77666		79302	2,106	78870	1,550
30_4_2_8_1p	72252		78265	8,322	76055	5,264
32_1_1_7	92865		92865	0,000	92865	0,000
32_1_1_8			88050		87137	
34_1_1_7	94843		1 i		98381	3,730
34_1_2_7	99426		1 i		105728	
34_1_2_7_1p	91206		1 i		93580	2,603
36_1_1_6	80991	99949	103403	27,672	99864	23,303
36_1_1_7			4 i		2,5 i	
36_1_2_1			3 i		2,2 i	
36_1_2_6			2 i		1,7 i	
36_1_2_8			2 i		1,6 i	
36_2_1_6	102881		1 i		106377	3,398
36_2_2_6	118495		122425	3,317	118495	0,000
36_2_2_6_1p			106832		105081	
36_2_2_6_2p	146481		149568	2,107	148967	1,697
36_2_2_6_92p	281421		308167	9,504	305453	8,540

Tabela 1 - continuação

Custos do ES1 para instâncias de 15 a 65 cidades [10 50]

Instância	Sarubbi, 2003		Pereira, 2004		ES1	
	Lim.Inf.	Lim.Sup.	10-50	%	10-50	%
36_2_2_6_9p	431450		431450	0,000	431450	0,000
36_2_2_6_mix	255472		255992	0,204	255472	0,000
38_25_1_6	115775		115775	0,000	115775	0,000
40_20_1_5	129514		129514	0,000	129514	0,000
40_20_1_6	124433		1 i		128070	2,923
40_25_1_6	111451	111458	1 i		112361	0,817
45_20_1_5	156871		156871	0,000	156871	0,000
50_1_1_5			178515		177389	
50_20_1_5	172417		183494	6,425	173554	0,659
55_40_1_5	209107		1 i		216734 (7)	3,647
55_45_1_5	205928		206536	0,295	205928	0,000
55_45_1_5_2p	305249		307972	0,892	306481	0,404
60_20_1_5	170547	224576	2 i		1,7 i	
60_30_1_4			1 i		1 i	
60_30_1_5	126939		1 i		149213	17,547
65_5_1_4	252003		2 i		1,3 i	
Totais			28,0	18	13,0	8
Médias			1,556		2,413	1,932

Com os parâmetros [20 10 100] (tabela 2) a factibilidade aumenta para 92,18%, caindo para cinco o número de instâncias inactíveis, sendo que a média percentual de resultados com relação à proximidade da solução exata cresce em 0,66% em relação aos parâmetros [10 50].

Tabela 2

Custos do ES1 para instâncias de 15 a 65 cidades [20 10 100]

Instância	Sarubbi, 2003		Pereira, 2004		ES1	
	Lim.Inf.	Lim.Sup.	20-10-100	%	20-10-100	%
15_4_25_25	128127		128127	0,000	128127	0,000
15_4_5_25	61515		61656	0,229	61515	0,000
15_4_5_25_1p	41481		41481	0,000	41481	0,000
15_7_1_15	33016		33016	0,000	33016	0,000
15_7_2_15	33892		33944	0,153	33892	0,000
18_7_1_17	37623		37623	0,000	37623	0,000
18_7_2_17	51869		51869	0,000	51869	0,000
18_7_3_15			2 i		53192	
18_7_3_17	47877		47877	0,000	47877	0,000
20_18_1_13			1 i		48876	
20_18_1_15	47450		47450	0,000	47450	0,000
20_18_3_15	53627	53630	53630	0,006	53627	0,000
20_18_3_15_1p	43970		43970	0,000	43970	0,000
20_18_5_15	67271		67271	0,000	67271	0,000
20_18_5_15_2p	51521		51521	0,000	51521	0,000

Tabela 2 - continuação
 Custos do ES1 para instâncias de 15 a 65 cidades [20 10 100]

Instância	Sarubbi, 2003		Pereira, 2004		ES1	
	Lim.Inf.	Lim.Sup.	20-10-100	%	20-10-100	%
22_10_1_13	53233		53233	0,000	53233	0,000
22_10_2_13	47330		47330	0,000	47330	0,000
22_10_2_13_1p	45250		45250	0,000	45250	0,000
25_17_1_11	53134		53134	0,000	53134	0,000
25_17_2_11	64262		64536	0,4264	64262	0,000
25_17_3_11	76400		1 i		78335	2,533
25_17_3_11_1p	63426		63426	0,000	63426	0,000
26_19_1_10	53358		53358	0,000	53358	0,000
26_19_1_13	54310	62616	62301	14,714	59461	9,484
28_20_1_10	71066		71066	0,000	71066	0,000
28_20_2_10	78054		78054	0,000	78054	0,000
28_20_2_10_1p	71964		1 i		74593	3,653
29_2_0_9	51128		51128	0,000	51128	0,000
29_2_1_9	79523		80372	1,068	79523	0,000
30_4_1_8	71198		71198	0,000	71198	0,000
30_4_1_8_1p			71198		70327	
30_4_2_8	77666		82028	5,616	79187	1,958
30_4_2_8_1p	72252		78265	8,322	78093	8,084
32_1_1_7	92865		92865	0,000	92865	0,000
32_1_1_8			84089		83122	
34_1_1_7	94843		97647	2,956	96837	2,102
34_1_2_7	99426		1 i		110749	11,388
34_1_2_7_1p	91206		91206	0,000	91206	0,000
36_1_1_6	80991	99949	103159	27,371	96058	18,603
36_1_1_7			3 i		2,3 i	
36_1_2_1			3 i		2,7 i	
36_1_2_6			2 i		1,4 i	
36_1_2_8			2 i		1,1 i	
36_2_1_6	102881		1 i		104258	1,338
36_2_2_6	118495		122425	3,317	118495	0,000
36_2_2_6_1p			106832		105730	
36_2_2_6_2p	146481		146481	0,000	146481	0,000
36_2_2_6_92p	281421		300041	6,616	291016	3,409
36_2_2_6_9p	431450		435125	0,852	431450	0,000
36_2_2_6_mix	255472		255472	0,000	255472	0,000
38_25_1_6	115775		1 i		117907	1,842
40_20_1_5	129514		129514	0,000	129514	0,000
40_20_1_6	124433		127807	2,711	126571	1,718
40_25_1_6	111451	111458	111458	0,006	111451	0,000
45_20_1_5	156871		1 i		156871	0,000
50_1_1_5			179982		178028	
50_20_1_5	172417		1 i		177332	2,851
55_40_1_5	209107		1 i		220348 (4)	5,376
55_45_1_5	205928		1 i		205928	0,000
55_45_1_5_2p	305249		1 i		308787	1,159
60_20_1_5	170547	224576	1 i		200413	17,512
60_30_1_4			2 i		1,3 i	
60_30_1_5	126939		1 i		183664	44,687

Tabela 2 - continuação

Custos do ES1 para instâncias de 15 a 65 cidades [20 10 100]

Instância	Sarubbi, 2003		Pereira, 2004		ES1	
	Lim.Inf.	Lim.Sup.	20-10-100	%	20-10-100	%
65_5_1_4	252003		252003	0,000	252003	0,000
Totais			27,0	19	8,8	5
Médias			1,421	1,814	1,760	2,598

Com os parâmetros [50 30 150] (tabela 3) a factibilidade aumenta mais ainda, chegando a 93,75%, reduzindo o número de instâncias inactíveis para quatro e a média percentual em relação à solução ótima ficou em 1,903%, atingindo a melhor média entre os três parâmetros.

Tabela 3

Custos do ES1 para instâncias de 15 a 65 cidades [50 30 150]

Instância	Sarubbi		Pereira, 2004		ES1	
	Lim.Inf.	Lim.Sup.	50-30-150	%	50-30-150	%
15_4_25_25	128127		128127	0,000	128127	0,000
15_4_5_25	61515		61656	0,229	61548	0,054
15_4_5_25_1p	41481		41481	0,000	41481	0,000
15_7_1_15	33016		33016	0,000	33016	0,000
15_7_2_15	33892		33944	0,153	33892	0,000
18_7_1_17	37623		37623	0,000	37623	0,000
18_7_2_17	51869		51869	0,000	51869	0,000
18_7_3_15			2 i		53897	
18_7_3_17	47877		47877	0,000	47877	0,000
20_18_1_13			2 i		49713	
20_18_1_15	47450		47450		47450	0,000
20_18_3_15	53627	53630	53630	0,006	53627	0,000
20_18_3_15_1p	43970		43970	0,000	43970	0,000
20_18_5_15	67271		67271	0,000	67271	0,000
20_18_5_15_2p	51521		51521	0,000	51521	0,000
22_10_1_13	53233		53233	0,000	53233	0,000
22_10_2_13	47330		47330	0,000	47330	0,000
22_10_2_13_1p	45250		45250	0,000	45250	0,000
25_17_1_11	53134		53134	0,000	53134	0,000
25_17_2_11	64262		1 i		66373	3,285
25_17_3_11	76400		76400	0,000	76400	0,000
25_17_3_11_1p	63426	63426	63426	0,000	63426	0,000
26_19_1_10	53358		53358	0,000	53358	0,000
26_19_1_13	54310	62616	59673	9,875	59977	10,435
28_20_1_10	71066		71066	0,000	71066	0,000
28_20_2_10	78054		78054	0,000	78054	0,000
28_20_2_10_1p	71964		71964	0,000	71964	0,000
29_2_0_9	51128		51128	0,000	51128	0,000
29_2_1_9	79523		79523	0,000	79523	0,000
30_4_1_8	71198		71198	0,000	71198	0,000

Tabela 3 - continuação

Custos do ES1 para instâncias de 15 a 65 cidades [50 30 150]

Instância	Sarubbi		Pereira, 2004		ES1	
	Lim.Inf.	Lim.Sup.	50-30-150	%	50-30-150	%
30_4_1_8_1p			71198		71083	
30_4_2_8	77666		79302	2,106	78200	0,688
30_4_2_8_1p	72252		73900	2,281	73739	2,058
32_1_1_7	92865		92865	0,000	92865	0,000
32_1_1_8			84530		82975	
34_1_1_7	94843		94843	0,000	94843	0,000
34_1_2_7	99426		1 i		106744	7,360
34_1_2_7_1p	91206		91206	0,000	91206	0,000
36_1_1_6	80991	99949	101731	25,608	86419	6,702
36_1_1_7			3 i		1,7 i	
36_1_2_1			3 i		2,3 i	
36_1_2_6			2 i		1,9 i	
36_1_2_8			1 i		101847 (3)	
36_2_1_6	102881		102881	0,000	102881	0,000
36_2_2_6	118495		118495	0,000	118495	0,000
36_2_2_6_1p			106832		106028	
36_2_2_6_2p	146481		146481	0,000	146481	0,000
36_2_2_6_92p	281421		300041	6,616	292776	4,035
36_2_2_6_9p	431450		431450	0,000	431450	0,000
36_2_2_6_mix	255472		255472	0,000	255472	0,000
38_25_1_6	115775		115775	0,000	115775	0,000
40_20_1_5	129514		129514	0,000	129514	0,000
40_20_1_6	124433		125217	0,630	124433	0,000
40_25_1_6	111451	111458	114853	3,052	114254	2,515
45_20_1_5	156871		1 i		157059	0,120
50_1_1_5			174832		172159	
50_20_1_5	172417		176446	2,337	174435	1,170
55_40_1_5	209107		1 i		217665	4,093
55_45_1_5	205928		1 i		205928	0,000
55_45_1_5_2p	305249		1 i		306028	0,255
60_20_1_5	170547	224576	1 i		203066	19,067
60_30_1_4			1 i		1 i	
60_30_1_5	126939		216471	70,532	176447	39,001
65_5_1_4	252003		1 i		252003	0,000
Totais			22,0	15	6,9	4
Médias			1,467		2,805	1,903

Com relação ao tempo computacional gasto pelo ES1, conforme a tabela 4, o mesmo obteve um desempenho superior em quase todas as instâncias em relação ao método exato. Em outra análise, percebe-se que os parâmetros [10 50] possuem a menor média de tempo, exatamente porque o espaço de busca conta com uma população de dez indivíduos apenas e o número de iterações é três vezes menor do que nos últimos parâmetros testados ([50 30 150]).

Todos os resultados encontrados através do ES1 dessa dissertação, tanto relativo a custos quanto a tempo são melhores do que os resultados de Pereira (2004). Comparando-se os resultados dos três parâmetros testados, nota-se que na medida em que aumenta o número de iterações, bem como o acréscimo da população, a solução vai degradando no que diz respeito a tempo computacional. Nesse sentido, a instância com 65 cidades poderá ser executada por um algoritmo exato, pois é melhor no tempo dispensado a sua execução.

Tabela 4

Tempo computacional para o ES1 com instâncias de 15 a 65 cidades

Instâncias	Sarubbi	10-50		20-10-100		50-30-150	
		Pereira, 2004	ES1	Pereira, 2004	ES1	Pereira, 2004	ES1
15_4_25_25	81,630	1,047	0,754	1,373	1,207	3,247	1,994
15_4_5_25	4,570	0,922	0,691	1,381	1,211	2,893	1,913
15_4_5_25_1p	4,060	0,784	0,622	1,449	1,245	2,781	1,891
15_7_1_15	8,550	0,925	0,693	1,360	1,198	2,908	2,022
15_7_2_15	1,760	1,065	0,763	1,531	1,262	3,150	1,967
18_7_1_17	51,180	1,545	1,003	3,284	2,061	7,707	4,257
18_7_2_17	5,440	1,878	1,169	2,944	1,992	5,149	3,075
18_7_3_15		2,059	1,260	2,778	1,909	6,593	3,797
18_7_3_17	10,940	1,454	0,957	3,467	2,254	4,499	2,750
20_18_1_13		2,490	1,475	4,994	3,017	10,642	5,821
20_18_1_15	5,810	3,597	2,029	5,114	3,077	8,702	4,851
20_18_3_15	539,490	3,111	1,786	4,591	2,799	8,207	4,688
20_18_3_15_1p	600,340	3,123	1,792	6,482	3,761	10,648	5,824
20_18_5_15	44,780	3,033	1,747	5,018	3,029	11,087	6,044
20_18_5_15_2p	28,840	2,484	1,472	5,450	3,245	9,657	5,329
22_10_1_13	65,830	4,616	2,538	7,109	4,076	12,761	6,754
22_10_2_13	28,270	3,364	1,912	5,287	3,164	13,149	7,075
22_10_2_13_1p	26,630	4,808	2,634	7,074	4,057	13,573	7,287
25_17_1_11	110,700	7,502	3,981	15,506	13,734	28,743	19,872
25_17_2_11	12,350	8,076	4,268	12,211	6,626	29,752	18,376
25_17_3_11	763,790	7,896	4,178	14,360	7,700	22,057	16,529
25_17_3_11_1p	629,560	8,613	4,537	16,329	9,685	32,540	18,770
26_19_1_10	102,900	10,169	5,315	20,796	15,161	38,478	19,739
26_19_1_13	8377,030	8,293	4,377	15,875	8,458	35,602	18,301
28_20_1_10	68,700	10,481	5,471	23,952	12,358	39,011	9,000
28_20_2_10	340,210	11,841	6,151	22,534	11,787	50,245	25,623
28_20_2_10_1p	336,690	13,390	6,925	21,925	11,483	46,385	23,693
29_2_0_9	2,540	18,825	9,643	21,991	11,516	55,399	28,200
29_2_1_9	266,890	12,375	6,418	29,050	15,045	63,873	32,437
30_4_1_8	96,810	20,438	10,548	32,132	20,577	71,649	36,325
30_4_1_8_1p		20,410	10,435	30,880	15,960	76,966	38,983
30_4_2_8	632,970	21,432	10,957	27,008	14,124	77,296	39,148
30_4_2_8_1p	796,880	16,172	8,316	27,501	14,271	49,793	25,397
32_1_1_7	190,340	19,098	9,779	32,147	22,800	60,049	30,525
32_1_1_8		22,190	11,325	41,324	21,182	76,262	38,631

Tabela 4 - continuação

Tempo computacional para o ES1 com instâncias de 15 a 65 cidades

Instâncias	Sarubbi	10-50		20-10-100		50-30-150	
		Pereira, 2004	ES1	Pereira, 2004	ES1	Pereira, 2004	ES1
34_1_1_7	190,340	28,583	14,522	57,391	29,216	117,607	59,304
34_1_2_7	137,290	61,894	31,177	69,545	36,223	157,218	79,109
34_1_2_7_1p	123,690	35,003	17,732	87,399	44,220	178,526	89,763
36_1_1_6	7200,650	42,530	21,495	61,793	31,417	125,860	63,430
36_1_1_7		134,313	67,387	183,130	92,085	464,236	232,618
36_1_2_1		89,260	44,860	165,377	83,209	371,346	186,173
36_1_2_6		67,088	33,774	84,769	42,905	196,088	98,544
36_1_2_8		124,966	72,937	115,725	58,383	386,140	193,570
36_2_1_6	33,920	71,849	36,155	104,476	52,758	189,711	95,356
36_2_2_6	429,290	57,393	28,927	75,708	38,374	135,633	68,317
36_2_2_6_1p		37,049	18,755	81,811	41,426	154,404	77,702
36_2_2_6_2p	507,210	56,683	28,572	75,316	38,178	176,170	88,585
36_2_2_6_92p	510,310	46,243	23,352	107,307	54,174	184,769	92,885
36_2_2_6_9p	480,550	57,068	28,764	79,797	40,419	184,331	92,666
36_2_2_6_mix	518,940	64,413	32,437	79,786	40,413	175,762	88,381
38_25_1_6	276,910	67,413	33,937	118,845	59,934	235,610	118,305
40_20_1_5	101,210	93,947	47,204	199,803	100,422	477,625	239,313
40_20_1_6	1294,610	105,111	52,786	138,845	69,943	436,774	218,887
40_25_1_6	6328,580	109,213	54,837	168,679	84,860	433,455	217,228
45_20_1_5	221,750	261,188	199,248	394,225	197,630	732,883	366,942
50_1_1_5		429,409	214,935	786,840	393,940	1696,872	848,936
50_20_1_5	2739,980	786,946	393,703	1258,141	629,593	2959,488	1480,244
55_40_1_5	2165,050	912,553	456,507	2100,070	1050,555	4143,910	2072,455
55_45_1_5	3339,890	1049,228	524,844	1839,791	998,416	4459,706	2230,353
55_45_1_5_2p	3774,870	729,525	364,993	1741,908	871,474	3826,086	1913,543
60_20_1_5	7202,450	5332,077	2666,269	5872,078	2936,559	13337,742	6669,371
60_30_1_4		2224,734	1112,597	5411,551	2706,296	10962,947	5481,974
60_30_1_5	7202,090	4057,066	3129,990	4903,793	2452,417	9503,401	4752,201
65_5_1_4	3074,040	6713,256	4277,332	10047,817	5024,429	27593,576	13797,215
Médias	1171,511	376,961	221,530	576,374	290,264	1327,958	664,535

5.2 Algoritmo ES2

A estratégia evolutiva ES2 utilizou os mesmos parâmetros de entrada para testes da estratégia ES1 e também o mesmo conjunto de instâncias.

Os parâmetros [10 50] (tabela 5) apresentaram uma factibilidade de 31,25% (28% maior que o trabalho de Pereira), diminui o número de arco fantasmas de 235 para 176,2, encontrando a solução ótima em apenas uma instância, fazendo com que a média de proximidade do ótimo fique em 11,268%.

Tabela 5

Custos do ES2 para instâncias de 15 a 65 cidades [10 50]

Instância	Sarubbi, 2003		ES2 Pereira, 2004		ES2	
	Lim.Inf.	Lim.Sup.	10-50	%	10-50	%
15_4_25_25	128127		1 i		132679	3,553
15_4_5_25	61515		1 i		64827	5,384
15_4_5_25_1p	41481		1 i		41481	0,000
15_7_1_15	33016		35257	6,788	34877	5,637
15_7_2_15	33892		1 i		36951	9,026
18_7_1_17	37623		49767	32,278	41400	10,039
18_7_2_17	51869		1 i		58903	13,561
18_7_3_15			2 i		1,3 i	
18_7_3_17	47877		1 i		55769	16,484
20_18_1_13			2 i		1,8 i	
20_18_1_15	47450		1 i		53028	11,756
20_18_3_15	53627	53630	1 i		63054	17,579
20_18_3_15_1p	43970		1 i		52467 (7)	19,325
20_18_5_15	67271		2 i		73583	9,383
20_18_5_15_2p	51521		2 i		61052	18,499
22_10_1_13	53233		2 i		61771	16,039
22_10_2_13	47330		2 i		54445	15,033
22_10_2_13_1p	45250		2 i		1,2 i	
25_17_1_11	53134		2 i		57881	8,934
25_17_2_11	64262		4 i		2,7 i	
25_17_3_11	76400		3 i		2,1 i	
25_17_3_11_1p	63426		1 i		69663	9,834
26_19_1_10	53358		2 i		60406	13,209
26_19_1_13	54310	62616	2 i		61238 (6)	12,756
28_20_1_10	71066		3 i		1,6 i	
28_20_2_10	78054		2 i		85337 (4)	9,331
28_20_2_10_1p	71964		3 i		2,8 i	
29_2_0_9	51128		2 i		2 i	
29_2_1_9	79523		3 i		1,9 i	
30_4_1_8	71198		3 i		2,3 i	
30_4_1_8_1p			3 i		2,3 i	
30_4_2_8	77666		3 i		2,5 i	
30_4_2_8_1p	72252		3 i		2,3 i	
32_1_1_7	92865		2 i		1,7 i	
32_1_1_8			3 i		2,5 i	
34_1_1_7	94843		4 i		3,1 i	
34_1_2_7	99426		3 i		2,9 i	
34_1_2_7_1p	91206		4 i		3,1 i	
36_1_1_6	80991	99949	3 i		2,8 i	
36_1_1_7			7 i		6,0 i	
36_1_2_1			6 i		5,1 i	
36_1_2_6			5 i		4,8 i	
36_1_2_8			5 i		4,8 i	
36_2_1_6	102881		4 i		3,7 i	
36_2_2_6	118495		6 i		5,8 i	
36_2_2_6_1p			5 i		4,0 i	
36_2_2_6_2p	146481		6 i		3,7 i	
36_2_2_6_92p	281421		4 i		2,4 i	

Tabela 5 - continuação
Custos do ES2 para instâncias de 15 a 65 cidades [10 50]

Instância	Sarubbi, 2003		ES2 Pereira, 2004		ES2	
	Lim.Inf.	Lim.Sup.	10-50	%	10-50	%
36_2_2_6_9p	431450		5 i		3,4 i	
36_2_2_6_mix	255472		3 i		2,8 i	
38_25_1_6	115775		5 i		4,2 i	
40_20_1_5	129514		5 i		4,2 i	
40_20_1_6	124433		5 i		4,3 i	
40_25_1_6	111451	111458	3 i		2,6 i	
45_20_1_5	156871		7 i		6,1 i	
50_1_1_5			7 i		6,1 i	
50_20_1_5	172417		8 i		7,3 i	
55_40_1_5	209107		7 i		6,9 i	
55_45_1_5	205928		8 i		7,4 i	
55_45_1_5_2p	305249		8 i		7,4 i	
60_20_1_5	170547	224576	9 i		7,9 i	
60_30_1_4			8 i		7,5 i	
60_30_1_5	126939		10 i		8,7 i	
65_5_1_4	252003		8 i		6,2 i	
Totais			235,0	62	176,2	44
Médias			3,790		19,533	11,268

Os testes realizados com os parâmetros [20 10 100] também demonstraram um aumento em relação à factibilidade dos testes de Pereira – 12,5% para 31,25% nesse trabalho, diminuíram o número de arcos fantasmas – de 212 para 156,9, e foi encontrada uma solução ótima, conforme a tabela 6.

Tabela 6
Custos do ES2 para instâncias de 15 a 65 cidades [20 10 100]

Instância	Sarubbi, 2003		Pereira, 2004		ES2	
	Lim.Inf.	Lim.Sup.	20-10-100	%	20-10-100	%
15_4_25_25	128127		156411	22,075	138518	8,110
15_4_5_25	61515		72867	18,454	67024	8,956
15_4_5_25_1p	41481		41481	0,000	41481	0,000
15_7_1_15	33016		36693	11,137	36233	9,744
15_7_2_15	33892		38846	14,617	36972	9,088
18_7_1_17	37623		44817	19,121	41811	11,131
18_7_2_17	51869		1		57845	11,521
18_7_3_15			2 i		1,1 i	
18_7_3_17	47877		1		56303	17,599
20_18_1_13			2 i		1,8 i	
20_18_1_15	47450		54197	14,219	51260	8,030
20_18_3_15	53627	53630	61452	14,592	58692	9,445
20_18_3_15_1p	43970		1 i		50356	14,524
20_18_5_15	67271		1 i		73952	9,931
20_18_5_15_2p	51521		1 i		55424	7,576
22_10_1_13	53233		1 i		55090	3,488

Tabela 6 - continuação
 Custos do ES2 para instâncias de 15 a 65 cidades [20 10 100]

Instância	Sarubbi, 2003		Pereira, 2004		ES2	
	Lim.Inf.	Lim.Sup.	20-10-100	%	20-10-100	%
22_10_2_13	47330		1 i		53086	12,161
22_10_2_13_1p	45250		2 i		1,3 i	
25_17_1_11	53134		2 i		62255	17,166
25_17_2_11	64262		2 i		1,7 i	
25_17_3_11	76400		2 i		1,5 i	
25_17_3_11_1p	63426		2 i		71681	13,015
26_19_1_10	53358		3 i		61438 (6)	15,143
26_19_1_13	54310	62616	1 i		58813	8,291
28_20_1_10	71066		2 i		79845	12,353
28_20_2_10	78054		3 i		1,2 i	
28_20_2_10_1p	71964		3 i		1,6 i	
29_2_0_9	51128		2 i		1,1 i	
29_2_1_9	79523		2 i		1,2 i	
30_4_1_8	71198		4 i		2,4 i	
30_4_1_8_1p			3 i		2,4 i	
30_4_2_8	77666		2 i		1,8 i	
30_4_2_8_1p	72252		2 i		1,6 i	
32_1_1_7	92865		3 i		2,2 i	
32_1_1_8			2 i		1,5 i	
34_1_1_7	94843		4 i		2,2 i	
34_1_2_7	99426		4 i		2,2 i	
34_1_2_7_1p	91206		2 i		1,1 i	
36_1_1_6	80991	99949	3 i		2,3 i	
36_1_1_7			7 i		5,8 i	
36_1_2_1			6 i		5,4 i	
36_1_2_6			6 i		5,4 i	
36_1_2_8			6 i		5,5 i	
36_2_1_6	102881		5 i		4,1 i	
36_2_2_6	118495		4 i		3,6 i	
36_2_2_6_1p			5 i		4,2 i	
36_2_2_6_2p	146481		3 i		2,3 i	
36_2_2_6_92p	281421		3 i		2,3 i	
36_2_2_6_9p	431450		4 i		2,9 i	
36_2_2_6_mix	255472		5 i		3,5 i	
38_25_1_6	115775		4 i		3,5 i	
40_20_1_5	129514		5 i		3,2 i	
40_20_1_6	124433		5 i		3,2 i	
40_25_1_6	111451	111458	4 i		3,3 i	
45_20_1_5	156871		5 i		4,8 i	
50_1_1_5			8 i		7,3 i	
50_20_1_5	172417		8 i		7,3 i	
55_40_1_5	209107		6 i		5,9 i	
55_45_1_5	205928		7 i		5,7 i	
55_45_1_5_2p	305249		8 i		6,6 i	
60_20_1_5	170547	224576	8 i		6,9 i	
60_30_1_4			7 i		6,7 i	
60_30_1_5	126939		9 i		7,8 i	
65_5_1_4	252003		8 i		7,5 i	

Tabela 6 - continuação

Custos do ES2 para instâncias de 15 a 65 cidades [20 10 100]

Totais	212	56		156,9	44	
Médias	3,786		14,277	3,566		10,364

Os últimos parâmetros a serem testados, [50 30 150], foram os quais apresentaram melhores resultados (tabela 7), principalmente por manter uma maior diversidade na população e por realizar um número bem maior de iterações do que os parâmetros anteriores.

Quanto à factibilidade a mesma ficou em 34,38%, melhorando em aproximadamente 23,50% em relação ao trabalho de Pereira. O número de arcos fantasmas foi o menor de todos os testados, ficando em 142,3, e foram encontradas duas soluções ótimas para duas instâncias de quinze cidades.

Nos resultados factíveis houve uma evolução no que diz respeito à proximidade da solução exata dos trabalhos testados, ficando com a melhor média de todos os parâmetros, 9,866%.

Tabela 7

Custos do ES2 para instâncias de 15 a 65 cidades [50 30 150]

Instância	Sarubbi		Pereira, 2004		ES2	
	Lim.Inf.	Lim.Sup.	50-30-150	%	50-30-150	%
15_4_25_25	128127		139715	9,044	134500	4,974
15_4_5_25	61515		61656	0,229	61515	0,000
15_4_5_25_1p	41481		41481	0,000	41481	0,000
15_7_1_15	33016		36868	11,667	35255	6,782
15_7_2_15	33892		1 i		37326	10,132
18_7_1_17	37623		44901	19,345	42095	11,886
18_7_2_17	51869		1 i		57901	11,629
18_7_3_15			2 i		63259 (5)	
18_7_3_17	47877		1 i		55582	16,093
20_18_1_13			2 i		1,2 i	
20_18_1_15	47450		1 i		50866	7,199
20_18_3_15	53627	53630	69946	30,431	62899	17,290
20_18_3_15_1p	43970		51688	17,553	49358	12,254
20_18_5_15	67271		1 i		72611	7,938
20_18_5_15_2p	51521		1 i		57998	12,572
22_10_1_13	53233		2 i		57550	8,110
22_10_2_13	47330		1 i		52045	9,962
22_10_2_13_1p	45250		2 i		1,8 i	
25_17_1_11	53134		2 i		59016	11,070
25_17_2_11	64262		2 i		1,2 i	
25_17_3_11	76400		2 i		1,1 i	
25_17_3_11_1p	63426	63426	1 i		68709	8,329
26_19_1_10	53358		2 i		59571	11,644

Tabela 7- continuação

Custos do ES2 para instâncias de 15 a 65 cidades [50 30 150]

Instância	Sarubbi		Pereira, 2004		ES2	
	Lim.Inf.	Lim.Sup.	50-30-150	%	50-30-150	%
26_19_1_13	54310	62616	2 i		61097 (8)	12,497
28_20_1_10	71066		3 i		81394	14,533
28_20_2_10	78054		2 i		87654	12,299
28_20_2_10_1p	71964		3 i		2,2 i	
29_2_0_9	51128		3 i		1,8 i	
29_2_1_9	79523		2 i		1,7 i	
30_4_1_8	71198		2 i		1,7 i	
30_4_1_8_1p			3 i		1,7 i	
30_4_2_8	77666		3 i		2,3 i	
30_4_2_8_1p	72252		2 i		1,6 i	
32_1_1_7	92865		3 i		1,8 i	
32_1_1_8			3 i		1,8 i	
34_1_1_7	94843		3 i		1,8 i	
34_1_2_7	99426		3 i		2,5 i	
34_1_2_7_1p	91206		4 i		3,1 i	
36_1_1_6	80991	99949	2 i		1,8 i	
36_1_1_7			6 i		4,6 i	
36_1_2_1			6 i		4,6 i	
36_1_2_6			5 i		4,4 i	
36_1_2_8			5 i		4,3 i	
36_2_1_6	102881		5 i		4,4 i	
36_2_2_6	118495		3 i		1,6 i	
36_2_2_6_1p			5 i		3,2 i	
36_2_2_6_2p	146481		4 i		3,1 i	
36_2_2_6_92p	281421		4 i		3,1 i	
36_2_2_6_9p	431450		4 i		3,2 i	
36_2_2_6_mix	255472		5 i		4,6 i	
38_25_1_6	115775		4 i		3,7 i	
40_20_1_5	129514		5 i		3,8 i	
40_20_1_6	124433		5 i		4,1 i	
40_25_1_6	111451	111458	3 i		2,2 i	
45_20_1_5	156871		7 i		5,6 i	
50_1_1_5			6 i		5,9 i	
50_20_1_5	172417		7 i		4,3 i	
55_40_1_5	209107		7 i		4,3 i	
55_45_1_5	205928		7 i		4,8 i	
55_45_1_5_2p	305249		6 i		4,8 i	
60_20_1_5	170547	224576	8 i		6,2 i	
60_30_1_4			7 i		6,1 i	
60_30_1_5	126939		8 i		7,2 i	
65_5_1_4	252003		8 i		7,1 i	
Totais			207	57	142,3	42
Médias			3,632		12,610	3,388
						9,866

O tempo de execução das instâncias testadas com o ES2 foi o melhor resultado encontrado em comparação com a estratégia evolutiva ES1. Traçando um

parâmetro do trabalho de Pereira com essa dissertação, pode-se constatar que houve um melhoramento em todas as instâncias testadas, conforme comprova a tabela 8, mostrada a seguir.

Tabela 8

Tempo computacional para o ES2 com instâncias de 15 a 65 cidades

Instâncias	Sarubbi	10-50		20-10-100		50-30-150	
		Pereira, 2004	ES2	Pereira, 2004	ES2	Pereira, 2004	ES1
15_4_25_25	81,630	1,047	0,754	1,373	1,207	3,247	1,994
15_4_5_25	4,570	0,922	0,691	1,381	1,211	2,893	1,913
15_4_5_25_1p	4,060	0,784	0,622	1,449	1,245	2,781	1,891
15_7_1_15	8,550	0,925	0,693	1,360	1,198	2,908	2,022
15_7_2_15	1,760	1,065	0,763	1,531	1,262	3,150	1,967
18_7_1_17	51,180	1,545	1,003	3,284	2,061	7,707	4,257
18_7_2_17	5,440	1,878	1,169	2,944	1,992	5,149	3,075
18_7_3_15		2,059	1,260	2,778	1,909	6,593	3,797
18_7_3_17	10,940	1,454	0,957	3,467	2,254	4,499	2,750
20_18_1_13		2,490	1,475	4,994	3,017	10,642	5,821
20_18_1_15	5,810	3,597	2,029	5,114	3,077	8,702	4,851
20_18_3_15	539,490	3,111	1,786	4,591	2,799	8,207	4,688
20_18_3_15_1p	600,340	3,123	1,792	6,482	3,761	10,648	5,824
20_18_5_15	44,780	3,033	1,747	5,018	3,029	11,087	6,044
20_18_5_15_2p	28,840	2,484	1,472	5,450	3,245	9,657	5,329
22_10_1_13	65,830	4,616	2,538	7,109	4,076	12,761	6,754
22_10_2_13	28,270	3,364	1,912	5,287	3,164	13,149	7,075
22_10_2_13_1p	26,630	4,808	2,634	7,074	4,057	13,573	7,287
25_17_1_11	110,700	7,502	3,981	15,506	13,734	28,743	19,872
25_17_2_11	12,350	8,076	4,268	12,211	6,626	29,752	18,376
25_17_3_11	763,790	7,896	4,178	14,360	7,700	22,057	16,529
25_17_3_11_1p	629,560	8,613	4,537	16,329	9,685	32,540	18,770
26_19_1_10	102,900	10,169	5,315	20,796	15,161	38,478	19,739
26_19_1_13	8377,030	8,293	4,377	15,875	8,458	35,602	18,301
28_20_1_10	68,700	10,481	5,471	23,952	12,358	39,011	9,000
28_20_2_10	340,210	11,841	6,151	22,534	11,787	50,245	25,623
28_20_2_10_1p	336,690	13,390	6,925	21,925	11,483	46,385	23,693
29_2_0_9	2,540	18,825	9,643	21,991	11,516	55,399	28,200
29_2_1_9	266,890	12,375	6,418	29,050	15,045	63,873	32,437
30_4_1_8	96,810	20,438	10,548	32,132	20,577	71,649	36,325
30_4_1_8_1p		20,410	10,435	30,880	15,960	76,966	38,983
30_4_2_8	632,970	21,432	10,957	27,008	14,124	77,296	39,148
30_4_2_8_1p	796,880	16,172	8,316	27,501	14,271	49,793	25,397
32_1_1_7	190,340	19,098	9,779	32,147	22,800	60,049	30,525
32_1_1_8		22,190	11,325	41,324	21,182	76,262	38,631
34_1_1_7	190,340	28,583	14,522	57,391	29,216	117,607	59,304
34_1_2_7	137,290	61,894	31,177	69,545	36,223	157,218	79,109
34_1_2_7_1p	123,690	35,003	17,732	87,399	44,220	178,526	89,763
36_1_1_6	7200,650	42,530	21,495	61,793	31,417	125,860	63,430
36_1_1_7		134,313	67,387	183,130	92,085	464,236	232,618
36_1_2_1		89,260	44,860	165,377	83,209	371,346	186,173

Tabela 8 - continuação

Tempo computacional para o ES2 com instâncias de 15 a 65 cidades

Instâncias	Sarubbi	10-50		20-10-100		50-30-150	
		Pereira, 2004	ES2	Pereira, 2004	ES2	Pereira, 2004	ES1
36_1_2_6		67,088	33,774	84,769	42,905	196,088	98,544
36_1_2_8		124,966	72,937	115,725	58,383	386,140	193,570
36_2_1_6	33,920	71,849	36,155	104,476	52,758	189,711	95,356
36_2_2_6	429,290	57,393	28,927	75,708	38,374	135,633	68,317
36_2_2_6_1p		37,049	18,755	81,811	41,426	154,404	77,702
36_2_2_6_2p	507,210	56,683	28,572	75,316	38,178	176,170	88,585
36_2_2_6_92p	510,310	46,243	23,352	107,307	54,174	184,769	92,885
36_2_2_6_9p	480,550	57,068	28,764	79,797	40,419	184,331	92,666
36_2_2_6_mix	518,940	64,413	32,437	79,786	40,413	175,762	88,381
38_25_1_6	276,910	67,413	33,937	118,845	59,934	235,610	118,305
40_20_1_5	101,210	93,947	47,204	199,803	100,422	477,625	239,313
40_20_1_6	1294,610	105,111	52,786	138,845	69,943	436,774	218,887
40_25_1_6	6328,580	109,213	54,837	168,679	84,860	433,455	217,228
45_20_1_5	221,750	261,188	199,248	394,225	197,630	732,883	366,942
50_1_1_5		429,409	214,935	786,840	393,940	1696,872	848,936
50_20_1_5	2739,980	786,946	393,703	1258,141	629,593	2959,488	1480,244
55_40_1_5	2165,050	912,553	456,507	2100,070	1050,555	4143,910	2072,455
55_45_1_5	3339,890	1049,228	524,844	1839,791	998,416	4459,706	2230,353
55_45_1_5_2p	3774,870	729,525	364,993	1741,908	871,474	3826,086	1913,543
60_20_1_5	7202,450	5332,077	2666,269	5872,078	2936,559	13337,742	6669,371
60_30_1_4		2224,734	1112,597	5411,551	2706,296	10962,947	5481,974
60_30_1_5	7202,090	4057,066	3129,990	4903,793	2452,417	9503,401	4752,201
65_5_1_4	3074,040	6713,256	4277,332	10047,817	5024,429	27593,576	13797,215
Médias	1171,511	376,961	221,530	576,374	290,264	1327,958	664,535

A estratégia evolutiva ES2 não conseguiu obter resultados factíveis em todas as suas instâncias. Isso se deve ao fato de que foram testadas com vértices pouco densos, que nesse caso são melhores tratados através de algoritmos exatos, pois quanto menos conexões, menos rotas a serem avaliadas.

5.3 Algoritmo ES3

A estratégia evolutiva ES3 também utilizou as mesmas instâncias das anteriores, porém as comparações foram realizadas com os resultados do método exato e com os resultados da estratégia ES1 desse trabalho.

Para os parâmetros [10-50] (tabela 9) houve uma piora na solução em 34 instâncias, no que diz respeito à proximidade da solução exata, sendo encontrado o ótimo em apenas 21,88% das instâncias. A factibilidade continua a mesma do ES1, porém aumenta em quatro o número de instâncias com arcos fantasmas.

Tabela 9

Custos do ES3 para instâncias de 15 a 65 cidades [10 50]

Instância	Sarubbi, 2003		10-50			
	Lim.Inf.	Lim.Sup.	ES1	%	ES3	%
15_4_25_25	128127		128127	0,000	128127	0,000
15_4_5_25	61515		61515	0,000	62791	2,074
15_4_5_25_1p	41481		41481	0,000	41481	0,000
15_7_1_15	33016		33016	0,000	33016	0,000
15_7_2_15	33892		33899	0,021	35463	4,635
18_7_1_17	37623		37623	0,000	37623	0,000
18_7_2_17	51869		51869	0,000	51869	0,000
18_7_3_15			(55367) (7)		1	
18_7_3_17	47877		47877	0,000	47877	0,000
20_18_1_13			1 i		1	
20_18_1_15	47450		47450	0,000	47450	0,000
20_18_3_15	53627	53630	53627	0,000	53712	0,159
20_18_3_15_1p	43970		44021	0,116	45028	2,406
20_18_5_15	67271		67271	0,000	67271	0,000
20_18_5_15_2p	51521		51521	0,000	51521	0,000
22_10_1_13	53233		53233	0,000	53297	0,120
22_10_2_13	47330		47330	0,000	47330	0,000
22_10_2_13_1p	45250		49871	10,212	50831	12,334
25_17_1_11	53134		53134	0,000	53134	0,000
25_17_2_11	64262		64262	0,000	66295	3,164
25_17_3_11	76400		77001	0,7866	77149	0,980
25_17_3_11_1p	63426		63426	0,000	65004	2,488
26_19_1_10	53358		53358	0,000	53823	0,871
26_19_1_13	54310	62616	56793	4,572	59738	9,994
28_20_1_10	71066		71066	0,000	72174	1,559
28_20_2_10	78054		78054	0,000	78054	0,000
28_20_2_10_1p	71964		75409	4,787	75677	5,160
29_2_0_9	51128		51128	0,000	53677	4,986
29_2_1_9	79523		79523	0,000	81025	1,889
30_4_1_8	71198		71198	0,000	73649	3,443
30_4_1_8_1p			70258		71643	
30_4_2_8	77666		78870	1,550	82961	6,818
30_4_2_8_1p	72252		76055	5,264	76842	6,353
32_1_1_7	92865		92865	0,000	94670	1,944
32_1_1_8			87137		88362	
34_1_1_7	94843		98381	3,730	101187	6,689
34_1_2_7	99426		102728		2,2	
34_1_2_7_1p	91206		93580	2,603	94553	3,670
36_1_1_6	80991	99949	99864	23,303	102249	26,247
36_1_1_7			2,5 i		2,6	
36_1_2_1			2,2 i		2,2	
36_1_2_6			1,7 i		1,8	
36_1_2_8			1,6 i		1,8	
36_2_1_6	102881		106377	3,398	106994	3,998
36_2_2_6	118495		118495	0,000	118495	0,000
36_2_2_6_1p			105081		108749	
36_2_2_6_2p	146481		148967	1,697	149451	2,028
36_2_2_6_92p	281421		305453	8,540	305996	8,732

Tabela 9 - continuação

Custos do ES3 para instâncias de 15 a 65 cidades [10 50]

Instância	Sarubbi, 2003		10-50			
	Lim.Inf.	Lim.Sup.	ES1	%	ES3	%
36_2_2_6_9p	431450		431450	0,000	431450	0,000
36_2_2_6_mix	255472		255472	0,000	257803	0,912
38_25_1_6	115775		115775	0,000	118043	1,959
40_20_1_5	129514		129514	0,000	131187	1,292
40_20_1_6	124433		128070	2,923	129539	4,103
40_25_1_6	111451	111458	112361	0,817	113081	1,463
45_20_1_5	156871		156871	0,000	158864	1,270
50_1_1_5			177389		181238 (4)	
50_20_1_5	172417		173554	0,659	175257	1,647
55_40_1_5	209107		216734 (7)	3,647	2,2	
55_45_1_5	205928		205928	0,000	207943	0,978
55_45_1_5_2p	305249		306481	0,404	310406	1,689
60_20_1_5	170547	224576	1,7 i		2,1	
60_30_1_4			1 i		1,8	
60_30_1_5	126939		149213 i	17,547	1,7	
65_5_1_4	252003		1,3 i		2,3	
Totais			13,0	8	21	12
Médias			1,625		1,750	2,876

Testando o algoritmo com os parâmetros [20 10 100], observa-se na tabela 10 que o algoritmo encontrou factibilidade em 22 instâncias, uma instância a menos do que o ES1, permanecendo com o mesmo percentual de otimalidade dos parâmetros [10 50] (tabela 9). O número de arcos fantasmas aumentou de oito para doze.

Tabela 10

Custos do ES3 para instâncias de 15 a 65 cidades [20 10 100]

Instância	Sarubbi, 2003		20-10-100			
	Lim.Inf.	Lim.Sup.	ES1	%	ES3	%
15_4_25_25	128127		128127	0,000	128127	0,000
15_4_5_25	61515		61515	0,000	61515	0,000
15_4_5_25_1p	41481		41481	0,000	41481	0,000
15_7_1_15	33016		33016	0,000	33016	0,000
15_7_2_15	33892		33892	0,000	34245	1,042
18_7_1_17	37623		37623	0,000	37623	0,000
18_7_2_17	51869		51869	0,000	51869	0,000
18_7_3_15			53192		56249 (7)	
18_7_3_17	47877		47877	0,000	47877	0,000
20_18_1_13			48876		51294	
20_18_1_15	47450		47450	0,000	47450	0,000
20_18_3_15	53627	53630	53627	0,000	53627	0,000
20_18_3_15_1p	43970		43970	0,000	43970	0,000
20_18_5_15	67271		67271	0,000	67271	0,000
20_18_5_15_2p	51521		51521	0,000	51521	0,000
22_10_1_13	53233		53233	0,000	54568	2,508

Tabela 10 – continuação
 Custos do ES3 para instâncias de 15 a 65 cidades [20 10 100]

22_10_2_13	47330		47330	0,000	47330		0,000
22_10_2_13_1p	45250		45250	0,000	46337		2,402
25_17_1_11	53134		53134	0,000	53134		0,000
25_17_2_11	64262		64262	0,000	65089		1,287
25_17_3_11	76400		78335	2,533	79101		3,535
25_17_3_11_1p	63426		63426	0,000	63426		0,000
26_19_1_10	53358		53358	0,000	54746		2,601
26_19_1_13	54310	62616	59461	9,484	60123		10,703
28_20_1_10	71066		71066	0,000	71066		0,000
28_20_2_10	78054		78054	0,000	78054		0,000
28_20_2_10_1p	71964		74593	3,653	74904		4,085
29_2_0_9	51128		51128	0,000	52434		2,554
29_2_1_9	79523		79523	0,000	79523		0,000
30_4_1_8	71198		71198	0,000	72668		2,065
30_4_1_8_1p			70327		71159		
30_4_2_8	77666		79187	1,958	81415		4,827
30_4_2_8_1p	72252		78093	8,084	78431		8,552
32_1_1_7	92865		92865	0,000	93576		0,766
32_1_1_8			83122		85794		
34_1_1_7	94843		96837	2,102	98477		3,832
34_1_2_7	99426		110749		112524	(3)	13,174
34_1_2_7_1p	91206		91206	0,000	92683		1,619
36_1_1_6	80991	99949	96058	18,603	100795		24,452
36_1_1_7			2,3 i		2,3 i		
36_1_2_1			2,7 i		2,8 i		
36_1_2_6			1,4 i		1,6 i		
36_1_2_8			1,1 i		1,4 i		
36_2_1_6	102881		104258	1,338	105338		2,388
36_2_2_6	118495		118495	0,000	118495		0,000
36_2_2_6_1p			105730		108041		
36_2_2_6_2p	146481		146481	0,000	147083		0,411
36_2_2_6_92p	281421		291016	3,409	295779		5,102
36_2_2_6_9p	431450		431450	0,000	431450		0,000
36_2_2_6_mix	255472		255472	0,000	256009		0,210
38_25_1_6	115775		117907	1,842	118164		2,063
40_20_1_5	129514		129514	0,000	130138		0,482
40_20_1_6	124433		126571	1,718	128296		3,104
40_25_1_6	111451	111458	111451	0,000	111451		0,000
45_20_1_5	156871		156871	0,000	157146		0,175
50_1_1_5			178028		181349		
50_20_1_5	172417		177332	2,851	178446		3,497
55_40_1_5	209107		220348	5,376	1,7 i	i	
55_45_1_5	205928		205928	0,000	205928		0,000
55_45_1_5_2p	305249		308787	1,159	309258		1,313
60_20_1_5	170547	224576	200413	17,512	206397		21,021
60_30_1_4			1,3 i		1,5 i	i	
60_30_1_5	126939		183664	44,687	1,3 i	i	
65_5_1_4	252003		252003	0,000	258162		2,444
Totais			8,8	5	12,6	7	
Médias			1,760	2,429	1,800		2,592

Os últimos parâmetros avaliados, [50 30 150], também não foram superiores ao ES1. Foram obtidos resultados factíveis em dezoito instâncias, e a otimalidade foi a melhor encontrada em relação ao próprio ES3, ficando em 28%, conforme a tabela 11 a seguir.

Tabela 11

Custos do ES3 para instâncias de 15 a 65 cidades [50 30 150]

Instância	Sarubbi		50-30-150			
	Lim.Inf.	Lim.Sup.	ES1	%	ES3	%
15_4_25_25	128127		128127	0,000	128127	0,000
15_4_5_25	61515		61548	0,054	61515	0,000
15_4_5_25_1p	41481		41481	0,000	41481	0,000
15_7_1_15	33016		33016	0,000	33016	0,000
15_7_2_15	33892		33892	0,000	33987	0,280
18_7_1_17	37623		37623	0,000	37623	0,000
18_7_2_17	51869		51869	0,000	51869	0,000
18_7_3_15			53897		56348	
18_7_3_17	47877		47877	0,000	47877	0,000
20_18_1_13			49713		50883	
20_18_1_15	47450		47450	0,000	47450	0,000
20_18_3_15	53627	53630	53627	0,000	53627	0,000
20_18_3_15_1p	43970		43970	0,000	43970	0,000
20_18_5_15	67271		67271	0,000	67271	0,000
20_18_5_15_2p	51521		51521	0,000	51521	0,000
22_10_1_13	53233		53233	0,000	53379	0,274
22_10_2_13	47330		47330	0,000	47330	0,000
22_10_2_13_1p	45250		45250	0,000	46258	2,228
25_17_1_11	53134		53134	0,000	53134	0,000
25_17_2_11	64262		66373	3,285	68024	5,854
25_17_3_11	76400		76400	0,000	78687	2,993
25_17_3_11_1p	63426	63426	63426	0,000	63426	0,000
26_19_1_10	53358		53358	0,000	55676	4,344
26_19_1_13	54310	62616	59977	10,435	62028	14,211
28_20_1_10	71066		71066	0,000	71066	0,000
28_20_2_10	78054		78054	0,000	78054	0,000
28_20_2_10_1p	71964		71964	0,000	72478	0,714
29_2_0_9	51128		51128	0,000	51566	0,857
29_2_1_9	79523		79523	0,000	79523	0,000
30_4_1_8	71198		71198	0,000	74116	4,098
30_4_1_8_1p			71083		72008	
30_4_2_8	77666		78200	0,688	79112	1,862
30_4_2_8_1p	72252		73739	2,058	74258	2,776
32_1_1_7	92865		92865	0,000	92865	0,000
32_1_1_8			82975		85046	
34_1_1_7	94843		94843	0,000	97783	3,100
34_1_2_7	99426		106744	7,360	108449	9,075
34_1_2_7_1p	91206		91206	0,000	92271	1,168
36_1_1_6	80991	99949	86419	6,702	95276	17,638
36_1_1_7			1,7 i		1,7 i	

Tabela 11 - continuação

Custos do ES3 para instâncias de 15 a 65 cidades [50 30 150]

Instância	Sarubbi		50-30-150			
	Lim.Inf.	Lim.Sup.	ES1	%	ES3	%
36_1_2_1			2,3 i		1,9 i	
36_1_2_6			1,9 i		1,7 i	
36_1_2_8			101847 (3)		1,5 i	
36_2_1_6	102881		102881	0,000	103037	0,152
36_2_2_6	118495		118495	0,000	118602	0,090
36_2_2_6_1p			106028		108532	
36_2_2_6_2p	146481		146481	0,000	147058	0,394
36_2_2_6_92p	281421		292776	4,035	295167	4,884
36_2_2_6_9p	431450		431450	0,000	440987	2,210
36_2_2_6_mix	255472		255472	0,000	255472	0,000
38_25_1_6	115775		115775	0,000	116038	0,227
40_20_1_5	129514		129514	0,000	129514	0,000
40_20_1_6	124433		124433	0,000	124433	0,000
40_25_1_6	111451	111458	114254	2,515	114184	2,452
45_20_1_5	156871		157059	0,120	159741	1,830
50_1_1_5			172159		175204	
50_20_1_5	172417		174435	1,170	174972	1,482
55_40_1_5	209107		217665	4,093	220744 (9)	5,565
55_45_1_5	205928		205928	0,000	205928	0,000
55_45_1_5_2p	305249		306028	0,255	307334	0,683
60_20_1_5	170547	224576	203066	19,067	206643	21,165
60_30_1_4			1		1,4	
60_30_1_5	126939		176447	39,001	179795	41,639
65_5_1_4	252003		252003	0,000	254577	1,021
Totais			6,9	4	8,2	5
Médias			1,725		1,640	2,930

O ES3 executa as mesmas instâncias em um tempo menor do que o ES1, podendo diminuir em aproximadamente três minutos a média de tempo. O mesmo comportamento apresentado pelo ES1 na instância de 65 cidades é verificado no ES3, ou seja, se o tempo for um aspecto fundamental nessas aplicações, nessa instância seria melhor executar o algoritmo exato. A tabela 12, mostrada a seguir, apresenta esses resultados.

Tabela 12

Tempo computacional para o ES3 com instâncias de 15 a 65 cidades

Instâncias	Sarubbi	10-50		20-10-100		50-30-150	
		ES1	ES3	ES1	ES3	ES1	ES3
15_4_25_25	81,630	0,754	0,714	1,207	0,898	1,994	1,361
15_4_5_25	4,570	0,691	0,654	1,211	0,901	1,913	1,306
15_4_5_25_1p	4,060	0,622	0,589	1,245	0,926	1,891	1,291
15_7_1_15	8,550	0,693	0,656	1,198	0,891	2,022	1,380
15_7_2_15	1,760	0,763	0,722	1,262	0,939	1,967	1,343

Tabela 12 – continuação

Tempo computacional para o ES3 com instâncias de 15 a 65 cidades

Instâncias	Sarubbi	10-50		20-10-100		50-30-150	
		ES1	ES3	ES1	ES3	ES1	ES3
18_7_1_17	51,180	1,003	0,949	2,061	1,533	4,257	2,906
18_7_2_17	5,440	1,169	1,107	1,992	1,482	3,075	2,099
18_7_3_15		1,260	1,193	1,909	1,420	3,797	2,592
18_7_3_17	10,940	0,957	0,906	2,254	1,676	2,750	1,877
20_18_1_13		1,475	1,397	3,017	2,244	5,821	3,974
20_18_1_15	5,810	2,029	1,921	3,077	2,289	4,851	3,312
20_18_3_15	539,490	1,786	1,691	2,799	2,082	4,688	3,200
20_18_3_15_1p	600,340	1,792	1,697	3,761	2,798	5,824	3,976
20_18_5_15	44,780	1,747	1,654	3,029	2,253	6,044	4,126
20_18_5_15_2p	28,840	1,472	1,394	3,245	2,414	5,329	3,638
22_10_1_13	65,830	2,538	2,404	4,076	3,032	6,754	4,611
22_10_2_13	28,270	1,912	1,811	3,164	2,353	7,075	4,829
22_10_2_13_1p	26,630	2,634	2,495	4,057	3,018	7,287	4,974
25_17_1_11	110,700	3,981	3,770	13,734	10,217	19,872	13,566
25_17_2_11	12,350	4,268	4,042	6,626	4,929	18,376	12,545
25_17_3_11	763,790	4,178	3,957	7,700	5,728	16,529	11,284
25_17_3_11_1p	629,560	4,537	4,297	9,685	7,205	18,770	12,814
26_19_1_10	102,900	5,315	5,033	15,161	11,279	19,739	13,475
26_19_1_13	8377,030	4,377	4,145	8,458	6,292	18,301	12,493
28_20_1_10	68,700	5,471	5,181	12,358	9,193	22,000	15,019
28_20_2_10	340,210	6,151	5,825	11,787	8,769	25,623	17,491
28_20_2_10_1p	336,690	6,925	6,559	11,483	8,542	23,693	16,174
29_2_0_9	2,540	9,643	9,132	11,516	10,567	28,200	19,251
29_2_1_9	266,890	6,418	6,078	15,045	11,192	32,437	22,143
30_4_1_8	96,810	10,548	9,990	20,577	15,308	36,325	24,797
30_4_1_8_1p		10,435	9,883	15,960	11,873	38,983	26,612
30_4_2_8	632,970	10,957	10,377	14,124	10,507	39,148	26,725
30_4_2_8_1p	796,880	8,316	7,876	14,271	10,616	25,397	17,337
32_1_1_7	190,340	9,779	9,262	22,800	16,961	30,525	20,838
32_1_1_8		11,325	10,726	21,182	15,758	38,631	26,372
34_1_1_7	190,340	14,522	13,753	29,216	21,734	59,304	40,484
34_1_2_7	137,290	31,177	29,528	36,223	26,947	79,109	54,005
34_1_2_7_1p	123,690	17,732	16,794	44,220	32,896	89,763	61,278
36_1_1_6	7200,650	21,495	20,358	31,417	23,371	63,430	43,301
36_1_1_7		67,387	63,822	92,085	68,504	232,618	158,799
36_1_2_1		44,860	42,487	83,209	61,901	186,173	127,093
36_1_2_6		33,774	31,987	42,905	36,918	98,544	67,272
36_1_2_8		72,937	69,079	58,383	43,432	193,570	132,142
36_2_1_6	33,920	36,155	34,242	52,758	39,248	95,356	65,095
36_2_2_6	429,290	28,927	27,396	38,374	30,547	68,317	46,637
36_2_2_6_1p		18,755	17,762	41,426	30,817	77,702	53,044
36_2_2_6_2p	507,210	28,572	27,060	38,178	31,401	88,585	60,473
36_2_2_6_92p	510,310	23,352	22,116	54,174	40,301	92,885	63,409
36_2_2_6_9p	480,550	28,764	27,242	40,419	30,068	92,666	63,259
36_2_2_6_mix	518,940	32,437	30,721	40,413	34,064	88,381	60,334
38_25_1_6	276,910	33,937	32,141	59,934	44,586	118,305	80,762
40_20_1_5	101,210	47,204	44,706	100,422	74,706	239,313	163,369
40_20_1_6	1294,610	52,786	49,993	69,943	55,032	218,887	149,425

Tabela 12 – continuação

Tempo computacional para o ES3 com instâncias de 15 a 65 cidades

Instâncias	Sarubbi	10-50		20-10-100		50-30-150	
		ES1	ES3	ES1	ES3	ES1	ES3
40_25_1_6	6328,580	54,837	51,936	84,860	63,129	217,228	148,293
45_20_1_5	221,750	199,248	188,708	197,630	195,749	366,942	250,496
50_1_1_5		214,935	203,564	393,940	293,061	848,936	579,535
50_20_1_5	2739,980	393,703	372,876	629,593	468,369	1480,244	1010,503
55_40_1_5	2165,050	456,507	432,357	1050,555	781,532	2072,455	1414,782
55_45_1_5	3339,890	524,844	497,080	998,416	742,745	2230,353	1522,573
55_45_1_5_2p	3774,870	364,993	345,684	871,474	648,310	1913,543	1306,299
60_20_1_5	7202,450	2666,269	2525,223	2936,559	2784,574	6669,371	4552,913
60_30_1_4		1112,597	1053,741	2706,296	2013,275	5481,974	3742,324
60_30_1_5	7202,090	3129,990	2964,414	4452,417	3312,255	4752,201	4044,137
65_5_1_4	3074,040	4277,332	4051,061	5024,429	4737,098	13797,215	9418,807
Médias	1171,511	221,530	209,811	321,514	265,229	664,738	466,290

O ES3 não conseguiu melhorar os resultados em termos de custos (função objetivo) em relação ao ES1, porém, como foi citado anteriormente, apresentou bons resultados quanto ao tempo de execução das instâncias.

5.4 Algoritmo ES4

Com as mesmas configurações dos algoritmos anteriores, a estratégia evolutiva ES4 obteve melhores resultados em todas as instâncias, comparando-se com o algoritmo ES2, visto que esse algoritmo apresenta um comportamento semelhante ao ES4, conforme seção 4.3.4.

Aplicando os parâmetros [10 50] (tabela 13), observa-se uma factibilidade de 48,44% (31 instâncias), melhorando em 17,19% em relação ao ES2, obtendo também somente uma solução ótima e diminuindo o número de arestas inactíveis de 44 para 33. A média da proximidade da solução ótima teve uma pequena variação, passando de 11,27% no ES2 para 10,80% no ES4.

Tabela 13

Custos do ES4 para instâncias de 15 a 65 cidades [10 50]

Instância	Sarubbi, 2003		ES2		ES4	
	Lim.Inf.	Lim.Sup.	10-50	%	10-50	%
15_4_25_25	128127		132679	3,553	129404	0,997
15_4_5_25	61515		64827	5,384	62021	0,823
15_4_5_25_1p	41481		41481	0,000	41481	0,000
15_7_1_15	33016		34877	5,637	34128	3,368
15_7_2_15	33892		36951	9,026	35699	5,332
18_7_1_17	37623		41400	10,039	40730	8,258

Tabela 13 - continuação

Custos do ES4 para instâncias de 15 a 65 cidades [10 50]

Instância	Sarubbi, 2003		ES2		ES4	
	Lim.Inf.	Lim.Sup.	10-50	%	10-50	%
18_7_2_17	51869		58903	13,561	58019	11,857
18_7_3_15			1,3 l		62185	
18_7_3_17	47877		55769	16,484	53966	12,718
20_18_1_13			1,8 l		1,3 i	
20_18_1_15	47450		53028	11,756	51537	8,613
20_18_3_15	53627	53630	63054	17,579	62430	16,415
20_18_3_15_1p	43970		52467	19,325	51928	19,325
20_18_5_15	67271		73583	9,383	72038	7,086
20_18_5_15_2p	51521		61052	18,499	58812	14,152
22_10_1_13	53233		61771	16,039	59267	11,335
22_10_2_13	47330		54445	15,033	54105	14,314
22_10_2_13_1p	45250		1,2 l		1,3 i	
25_17_1_11	53134		57881	8,934	57520	8,255
25_17_2_11	64262		2,7 l		2,2 i	
25_17_3_11	76400		2,1 l		1,4 i	
25_17_3_11_1p	63426		69663	9,834	68487	7,979
26_19_1_10	53358		60406	13,209	58731	10,070
26_19_1_13	54310	62616	61238	12,756	60692	11,751
28_20_1_10	71066		1,6 l		80314 (6)	13,013
28_20_2_10	78054		85337	9,331	85059	8,975
28_20_2_10_1p	71964		2,8 l		2,8 i	
29_2_0_9	51128		2 l		57658 (4)	12,772
29_2_1_9	79523		1,9 l		1,4 i	
30_4_1_8	71198		2,3 l		2 i	
30_4_1_8_1p			2,3 l		1,6 i	
30_4_2_8	77666		2,5 l		1,8 i	
30_4_2_8_1p	72252		2,3 l		1,5 i	
32_1_1_7	92865		1,7 l		105137	13,215
32_1_1_8			2,5 l		2,2 i	
34_1_1_7	94843		3,1 l		1,7 i	
34_1_2_7	99426		2,9 l		110293	10,930
34_1_2_7_1p	91206		3,1 l		2,7 i	
36_1_1_6	80991	99949	2,8 l		101955	25,884
36_1_1_7			6,0 l		3,5 i	
36_1_2_1			5,1 l		2,2 i	
36_1_2_6			4,8 l		3,8 i	
36_1_2_8			4,8 l		106731 i	
36_2_1_6	102881		3,7 l		115732 (8)	12,491
36_2_2_6	118495		5,8 l		4,7 i	
36_2_2_6_1p			4,0 l		3,5 i	
36_2_2_6_2p	146481		3,7 l		2,6 i	
36_2_2_6_92p	281421		2,4 l		307198	9,160
36_2_2_6_9p	431450		3,4 l		467023	8,245
36_2_2_6_mix	255472		2,8 l		321854 (7)	25,984
38_25_1_6	115775		4,2 l		2,2 i	
40_20_1_5	129514		4,2 l		3,0 i	
40_20_1_6	124433		4,3 l		3,0 i	
40_25_1_6	111451	111458	2,6 l		1,8 i	

Tabela 13 - continuação

Custos do ES4 para instâncias de 15 a 65 cidades [10 50]

Instância	Sarubbi, 2003		ES2		ES4	
	Lim.Inf.	Lim.Sup.	10-50	%	10-50	%
45_20_1_5	156871		6,1 l		5,3 i	
50_1_1_5			6,1 l		5,5 i	
50_20_1_5	172417		7,3 l		5,4 i	
55_40_1_5	209107		6,9 l		5,1 i	
55_45_1_5	205928		7,4 l		6,7 i	
55_45_1_5_2p	305249		7,4 l		6,7 i	
60_20_1_5	170547	224576	7,9 l		7,4 i	
60_30_1_4			7,5 l		7,5 i	
60_30_1_5	126939		8,7 l		7,5 i	
65_5_1_4	252003		6,2 l		7,1 i	
Totais			176,2	44	118,4	33
Médias			4,005		3,588	10,804

Os testes realizados com os parâmetros [20 10 100] demonstraram um aumento em relação à factibilidade dos testes do ES2, passando de 31,25% neste, para 60,94% no ES4, conforme tabela 14. Consequentemente diminui o número de arestas inactíveis – de 44 no ES2 para 25 no ES4. Ambos encontraram somente uma solução ótima. Em relação à proximidade da solução ótima houve uma melhora de 1,62%.

Tabela 14

Custos do ES4 para instâncias de 15 a 65 cidades [20 10 100]

Instância	Sarubbi, 2003		ES2		ES4	
	Lim.Inf.	Lim.Sup.	20-10-100	%	20-10-100	%
15_4_25_25	128127		138518	8,110	130541	1,884
15_4_5_25	61515		67024	8,956	63190	2,723
15_4_5_25_1p	41481		41481	0,000	41481	0,000
15_7_1_15	33016		36233	9,744	34773	5,322
15_7_2_15	33892		36972	9,088	34252	1,062
18_7_1_17	37623		41811	11,131	39109	3,950
18_7_2_17	51869		57845	11,521	56874	9,649
18_7_3_15			1,1 i		62320	
18_7_3_17	47877		56303	17,599	52555	9,771
20_18_1_13			1,8 i		63977	
20_18_1_15	47450		51260	8,030	48769	2,780
20_18_3_15	53627	53630	58692	9,445	57441	7,112
20_18_3_15_1p	43970		50356	14,524	50028	13,778
20_18_5_15	67271		73952	9,931	71599	6,434
20_18_5_15_2p	51521		55424	7,576	53653	4,138
22_10_1_13	53233		55090	3,488	54383	2,160
22_10_2_13	47330		53086	12,161	51572	8,963

Os últimos parâmetros a serem testados, [50 30 150], apresentaram uma factibilidade de 62,5%, melhorando 28,12% em relação ao ES2. O número de arcos fantasmas foi o menor de todos os testados, ficando em 24 arcos, e foram encontradas duas soluções ótimas entre todas as instâncias, mesmo número do ES2.

Nos resultados factíveis houve uma evolução no que diz respeito à proximidade da solução exata dos trabalhos testados, caindo de 9,87% no ES2 para 8,81% no ES4, como se pode observar na tabela 15.

Tabela 15

Custos do ES4 para instâncias de 15 a 65 cidades [50 30 150]

Instância	Sarubbi		ES2		ES4	
	Lim.Inf.	Lim.Sup.	50-30-150	%	50-30-150	%
15_4_25_25	128127		134500	4,974	130737	2,037
15_4_5_25	61515		61515	0,000	61515	0,000
15_4_5_25_1p	41481		41481	0,000	41481	0,000
15_7_1_15	33016		35255	6,782	34317	3,941
15_7_2_15	33892		37326	10,132	35307	4,175
18_7_1_17	37623		42095	11,886	40741	8,287
18_7_2_17	51869		57901	11,629	55685	7,357
18_7_3_15			63259 (5)		63020	
18_7_3_17	47877		55582	16,093	50936	6,389
20_18_1_13			1,2 i		62696	
20_18_1_15	47450		50866	7,199	48098	1,366
20_18_3_15	53627	53630	62899	17,290	60733	13,251
20_18_3_15_1p	43970		49358	12,254	48400	10,075
20_18_5_15	67271		72611	7,938	69546	3,382
20_18_5_15_2p	51521		57998	12,572	54193	5,186
22_10_1_13	53233		57550	8,110	55842	4,901
22_10_2_13	47330		52045	9,962	50149	5,956
22_10_2_13_1p	45250		1,8 i		56472 (8)	24,800
25_17_1_11	53134		59016	11,070	57768	8,721
25_17_2_11	64262		1,2 i		76742 (6)	19,420
25_17_3_11	76400		1,1 i		85038	11,306
25_17_3_11_1p	63426	63426	68709	8,329	66146	4,288
26_19_1_10	53358		59571	11,644	56263	5,444
26_19_1_13	54310	62616	61097 (8)	12,497	58176	12,497
28_20_1_10	71066		81394	14,533	80355	13,071
28_20_2_10	78054		87654	12,299	84459	8,206
28_20_2_10_1p	71964		2,2 i		1,1	
29_2_0_9	51128		1,8 i		56937	11,362
29_2_1_9	79523		1,7 i		87404	9,910
30_4_1_8	71198		1,7 i		79713	11,960
30_4_1_8_1p			1,7 i		1,3	
30_4_2_8	77666		2,3 i		1,7	
30_4_2_8_1p	72252		1,6 i		85008	17,655
32_1_1_7	92865		1,8 i		101383	9,172

Tabela 15 - continuação

Custos do ES4 para instâncias de 15 a 65 cidades [50 30 150]

Instância	Sarubbi		ES2		ES4	
	Lim.Inf.	Lim.Sup.	50-30-150	%	50-30-150	%
32_1_1_8			1,8 i		1,2	
34_1_1_7	94843		1,8 i		1,3	
34_1_2_7	99426		2,5 i		105948	6,560
34_1_2_7_1p	91206		3,1 i		1,6	
36_1_1_6	80991	99949	1,8 i		92115	13,735
36_1_1_7			4,6 i		2,9	
36_1_2_1			4,6 i		1,7	
36_1_2_6			4,4 i		2,3	
36_1_2_8			4,3 i		102840	
36_2_1_6	102881		4,4 i		113582	10,401
36_2_2_6	118495		1,6 i		128944	8,818
36_2_2_6_1p			3,2 i		1,9	
36_2_2_6_2p	146481		3,1 i		159111	8,622
36_2_2_6_92p	281421		3,1 i		298162	5,949
36_2_2_6_9p	431450		3,2 i		452133	4,794
36_2_2_6_mix	255472		4,6 i		314826	23,233
38_25_1_6	115775		3,7 i		1,7	
40_20_1_5	129514		3,8 i		2,8	
40_20_1_6	124433		4,1 i		2,7	
40_25_1_6	111451	111458	2,2 i		1,8	
45_20_1_5	156871		5,6 i		4,5	
50_1_1_5			5,9 i		3,8	
50_20_1_5	172417		4,3 i		3,1	
55_40_1_5	209107		4,3 i		2,7	
55_45_1_5	205928		4,8 i		4,1	
55_45_1_5_2p	305249		4,8 i		3,9	
60_20_1_5	170547	224576	6,2 i		5,5	
60_30_1_4			6,1 i		3,4	
60_30_1_5	126939		7,2 i		3,9	
65_5_1_4	252003		7,1 i		5,3	
Totais			142,3	42	66,2	24
Médias			3,388		9,866	2,758
						8,817

O tempo de execução das instâncias testadas com o ES4 foi superior em comparação com a estratégia evolutiva ES2, pelo fato de que na estratégia ES4 esse algoritmo realiza mais trocas de vértices, conforme comprova a tabela 16, mostrada a seguir.

Tabela 16

Tempo computacional para o ES4 com instâncias de 15 a 65 cidades

Instâncias	Sarubbi	ES4		
		10-50	20-10-100	50-30-150
15_4_25_25	81,630	0,461	0,848	1,442
15_4_5_25	4,570	0,423	0,851	1,383
15_4_5_25_1p	4,060	0,381	0,875	1,367
15_7_1_15	8,550	0,424	0,842	1,462
15_7_2_15	1,760	0,467	0,887	1,422
18_7_1_17	51,180	0,614	1,449	3,078
18_7_2_17	5,440	0,715	1,400	2,223
18_7_3_15		0,771	1,342	2,745
18_7_3_17	10,940	0,586	1,584	1,988
20_18_1_13		0,903	2,121	4,209
20_18_1_15	5,810	1,241	2,163	3,507
20_18_3_15	539,490	1,093	1,968	3,389
20_18_3_15_1p	600,340	1,096	2,644	4,211
20_18_5_15	44,780	1,069	2,129	4,369
20_18_5_15_2p	28,840	0,901	2,281	3,853
22_10_1_13	65,830	1,553	2,865	4,883
22_10_2_13	28,270	1,170	2,224	5,115
22_10_2_13_1p	26,630	1,612	2,852	5,268
25_17_1_11	110,700	2,436	9,655	14,367
25_17_2_11	12,350	2,612	4,658	13,286
25_17_3_11	763,790	2,557	5,413	11,950
25_17_3_11_1p	629,560	2,776	6,809	13,571
26_19_1_10	102,900	3,252	10,658	14,271
26_19_1_13	8377,030	2,678	5,946	13,232
28_20_1_10	68,700	3,348	8,688	6,507
28_20_2_10	340,210	3,764	8,286	18,525
28_20_2_10_1p	336,690	4,238	8,072	17,130
29_2_0_9	2,540	5,901	8,095	20,388
29_2_1_9	266,890	3,928	10,577	23,452
30_4_1_8	96,810	6,455	14,466	26,263
30_4_1_8_1p		6,386	11,220	28,185
30_4_2_8	632,970	6,706	9,929	28,304
30_4_2_8_1p	796,880	5,089	10,032	18,362
32_1_1_7	190,340	5,985	16,028	22,069
32_1_1_8		6,931	14,891	27,930
34_1_1_7	190,340	8,887	20,538	42,876
34_1_2_7	137,290	19,080	25,465	57,196
34_1_2_7_1p	123,690	10,852	31,086	64,899
36_1_1_6	7200,650	13,155	22,086	45,860
36_1_1_7		41,241	64,736	168,183
36_1_2_1		27,454	58,496	134,603
36_1_2_6		20,670	30,162	71,247
36_1_2_8		44,637	41,043	139,951
36_2_1_6	33,920	22,127	37,089	68,942
36_2_2_6	429,290	17,703	26,977	49,393
36_2_2_6_1p		11,478	29,122	56,179
36_2_2_6_2p	507,210	17,486	26,839	64,047
36_2_2_6_92p	510,310	14,291	38,084	67,155

Tabela 16 - continuação

Tempo computacional para o ES4 com instâncias de 15 a 65 cidades

Instâncias	Sarubbi	ES4		
		10-50	20-10-100	50-30-150
36_2_2_6_9p	480,550	17,604	28,414	66,997
36_2_2_6_mix	518,940	19,851	28,410	63,899
38_25_1_6	276,910	20,769	42,134	85,535
40_20_1_5	101,210	28,889	70,596	173,023
40_20_1_6	1294,610	32,305	49,170	158,255
40_25_1_6	6328,580	33,560	59,656	157,055
45_20_1_5	221,750	121,940	138,934	265,299
50_1_1_5		131,540	276,940	613,781
50_20_1_5	2739,980	240,946	442,604	1070,216
55_40_1_5	2165,050	279,382	738,540	1498,385
55_45_1_5	3339,890	321,205	701,886	1612,545
55_45_1_5_2p	3774,870	223,375	612,646	1383,492
60_20_1_5	7202,450	1631,756	2064,401	4821,955
60_30_1_4		680,909	1902,526	3963,467
60_30_1_5	7202,090	1915,554	1724,049	3435,841
65_5_1_4	3074,040	2617,727	3532,173	9975,386
Médias	1171,511	135,576	204,055	480,459

A estratégia evolutiva ES4 também não conseguiu obter resultados factíveis em todas as suas instâncias, porém apresentou melhora frente à estratégia ES2 em quase todos os aspectos com relação aos custos: menor número de arcos fantasmas, maior factibilidade e maior proximidade da solução exata, ficando com o as mesmas duas soluções ótimas da estratégia ES2.

5.5 Algoritmos Evolutivos com Alta Densidade

O algoritmo exato desenvolvido por Sarubbi (2003) não conseguiu obter a solução ótima em nenhuma instância com o mesmo número de cidades, mas com 70% de conectividade em todas elas. Esse fato se deve a necessidade do algoritmo exato ter que testar um número muito maior de possíveis soluções do que no grafo de baixa densidade.

Como o algoritmo exato não obteve resultados ótimos, serão apresentados nessa seção os resultados encontrados com os algoritmos evolutivos propostos nesse trabalho (tabela 17), criando um novo parâmetro para teste: [30 10 100].

Tabela 17: custos para instâncias de 15 a 65 cidades [30 10 100]

Instâncias	30-10-100			
	ES1	ES2	ES3	ES4
15_4_5_70	81926	82930	83774	81561
15_4_25_70	28313	29201	29113	27994
15_7_1_70	22867	25934	24102	22662
15_7_2_70	23497	24497	23975	22893
18_7_1_70	28228	31087	29102	29051
18_7_2_70	28165	28522	30107	27837
18_7_3_70	32166	33193	33820	31158
20_18_1_70	29957	30574	30567	30073
20_18_3_70	34332	34776	36030	33764
20_18_5_70	41680	42589	42529	42179
22_10_1_70	34670	36519	35376	33691
22_10_2_70	34347	36384	35046	34589
25_17_1_70	40407	43781	41229	41027
25_17_2_70	41491	43749	42336	42975
25_17_3_70	46644	48335	47593	47387
26_19_1_70	42383	42890	43245	42794
28_20_1_70	49263	50023	51787	49342
28_20_2_70	51103	51730	52143	51484
29_2_1_70	50591	52738	52620	51156
30_4_1_70	50818	53741	52852	51390
30_4_2_70	53400	55002	54874	54011
32_1_1_70	58436	62189	60625	59447
34_1_1_70	57708	63439	60882	59493
34_1_2_70	61861	64717	63120	62775
36_1_1_70	64895	67842	66216	66103
36_1_2_70	66609	70607	67965	66970
36_2_1_70	65887	67863	67227	66925
36_2_2_70	67788	70821	69167	68885
38_25_1_70	68037	71078	70922	69734
40_20_1_70	69567	75654	72982	70719
40_25_1_70	71163	73298	72611	71265
45_20_1_70	85065	89617	87796	85697
50_1_1_70	100884	106910	102937	102005
50_20_1_70	99350	107331	101372	100424
55_45_1_70	113289	121688	116594	115794
60_20_1_70	126305	140094	130876	128213
60_30_1_70	129131	143469	134759	132126
65_5_1_70	157983	190084	161198	159871
Médias	60795	64866	62618	61460

Todos os algoritmos evolutivos encontraram soluções factíveis em todas as instâncias de alta densidade, comprovando que para a resolução de grafos bastante densos é recomendado o uso de heurísticas e metaheurísticas na busca por soluções factíveis.

Pode-se notar também, conforme a tabela 18, que o tempo de execução das instâncias é muito inferior ao tempo gasto pelo algoritmo exato de Sarubbi (com as instâncias de baixa densidade).

Tabela 18: tempo computacional para instâncias de 15 a 65 cidades [30 10 100]

Instâncias	30-10-100			
	ES1	ES2	ES3	ES4
15_4_5_70	0,378	0,381	0,375	0,383
15_4_25_70	0,308	0,311	0,306	0,312
15_7_1_70	0,308	0,311	0,306	0,312
15_7_2_70	0,305	0,307	0,302	0,309
18_7_1_70	0,422	0,425	0,419	0,427
18_7_2_70	0,421	0,424	0,418	0,426
18_7_3_70	0,429	0,432	0,425	0,434
20_18_1_70	0,530	0,534	0,526	0,536
20_18_3_70	0,530	0,534	0,526	0,536
20_18_5_70	0,539	0,543	0,535	0,546
22_10_1_70	0,666	0,671	0,661	0,675
22_10_2_70	0,674	0,679	0,669	0,683
25_17_1_70	0,923	0,931	0,916	0,935
25_17_2_70	0,992	1,000	0,985	1,005
25_17_3_70	0,989	0,997	0,982	1,002
26_19_1_70	1,027	1,035	1,019	1,040
28_20_1_70	1,282	1,292	1,272	1,299
28_20_2_70	1,296	1,306	1,286	1,313
29_2_1_70	1,431	1,442	1,420	1,449
30_4_1_70	1,606	1,619	1,594	1,627
30_4_2_70	1,621	1,633	1,608	1,642
32_1_1_70	2,124	2,141	2,108	2,152
34_1_1_70	2,517	2,536	2,498	2,550
34_1_2_70	2,527	2,546	2,507	2,559
36_1_1_70	3,032	3,055	3,009	3,071
36_1_2_70	2,980	3,003	2,957	3,018
36_2_1_70	3,043	3,066	3,020	3,082
36_2_2_70	3,002	3,025	2,979	3,041
38_25_1_70	3,679	3,707	3,651	3,727
40_20_1_70	4,482	4,517	4,448	4,540
40_25_1_70	4,428	4,462	4,394	4,485
45_20_1_70	7,702	7,761	7,643	7,801
50_1_1_70	11,351	11,438	11,264	12,047
50_20_1_70	11,538	11,626	11,450	12,587
55_45_1_70	18,480	18,622	18,339	19,018
60_20_1_70	53,384	53,794	52,975	54,072
60_30_1_70	55,025	55,447	54,603	56,734
65_5_1_70	52,513	52,916	52,111	53,590
Média	6,802	6,854	6,750	6,973

6 CONCLUSÃO

O objetivo deste trabalho foi desenvolver um algoritmo evolutivo para a resolução do Problema do Caixeiro Viajante com Demandas Heterogêneas.

Em um primeiro momento foi realizada uma revisão bibliográfica da área de otimização combinatória, teoria da complexidade, algoritmos heurísticos e sobre o Problema do Caixeiro Viajante e sua extensão: o Problema do Caixeiro Viajante com Demandas Heterogêneas. Foram abordados também trabalhos relacionados ao tema da dissertação.

Após foram apresentados os algoritmos evolutivos com uma conceituação sobre cada um deles. Na seqüência discutiu-se sobre os métodos de resolução para o PCVDH.

As estratégias evolutivas ES1, ES2 , ES3 e ES4 foram os algoritmos propostos para a resolução do problema em questão. Essas quatro estratégias foram testadas com um conjunto de 64 instâncias de baixa densidade, para que se pudessem comparar seus resultados com os resultados obtidos pelo método exato de Sarubbi (2003) e com os algoritmos evolutivos de Pereira (2004), sendo que ainda foram apresentados resultados com a execução de 38 instâncias de baixa densidade, para as quais não existem soluções ótimas para comparação.

A estratégia evolutiva ES1 apresentou os melhores resultados em relação à função objetivo (custos) e às soluções factíveis, porém seu tempo de execução foi superior ao algoritmo exato em instâncias com mais de 45 cidades, com uma população inicial de cinquenta indivíduos. O ES1 conseguiu igualar ou melhorar os resultados em relação ao trabalho de Pereira (2004), tanto no que se refere a custos quanto a tempo de execução.

A estratégia evolutiva ES2 encontrou o ótimo em duas instâncias, sendo ainda o algoritmo que apresentou o menor número de soluções factíveis e o maior número de arcos fantasmas. Mesmo assim, essa metaheurística melhorou os resultados em relação à proximidade da solução exata, comparando-se com o trabalho de Pereira (2004). O tempo de execução do ES2 foi o melhor dos quatro algoritmos testados.

A estratégia evolutiva ES3 obteve melhores resultados em tempo de execução em relação ao ES1, principalmente porque uma operação com o método *swap* é bem menos complexa que uma operação k-opt. Quanto aos custos obtidos a estratégia ES1 foi superior em todas as instâncias testadas.

Por fim, foram testadas 38 instâncias de alta densidade, sendo que o algoritmo desenvolvido por Sarubbi não encontrou nenhuma solução ótima para elas, comprovando que o método exato somente consegue resolver problemas com vértices de baixa conectividade, justificando-se também o uso de metaheurísticas para os problemas com grafos altamente conexos.

Para trabalhos futuros podem ser testados outros métodos de resolução para o PCVDH, como por exemplo, a aplicação do algoritmo GENIUS (GENDREAU; HERTZ; LAPORTE, 1992) para uma tentativa de melhora na construção da solução inicial, permitindo que os métodos de buscas locais encontrem custos mais próximos da solução ótima, principalmente com a estratégia evolutiva ES2. Podem ser testados também outros parâmetros de configuração para as instâncias de cidades, variando mais o tamanho da população, a taxa de crescimento e o número de iterações, além do que existem outras técnicas de melhoramento na literatura que podem ser aplicadas.

Para a estratégia evolutivas ES1 podem ser adicionados outros algoritmos de busca dentro do conjunto de melhoramentos *MoveSet*. Com relação à estratégia evolutiva ES2 podem ser aplicados outros operadores de *crossover* e de mutação encontrados na literatura da área de otimização combinatória.

Em relação a estratégia evolutiva ES4 pode ser feita um alteração em seu algoritmo (seção 4.3.3): antes de testar o critério de parada, seleciona-se um dos indivíduos sorteados (P1, P2 ou P3) e aplica-se o parâmetro NMoves, inserindo o novo indivíduo na população, dessa forma se obtém uma maior diversidade da mesma, podendo então ser encontrados melhores resultados para os custos.

7 REFERÊNCIAS BIBLIOGRÁFICAS

ABLAY, P. *Optimieren mit Evolutionsstrategien*. Doktorarbeit, Fachbereich Wirtschafts- und Sozialwissenschaften, Universität Heidelberg, Heidelberg, 1979.

AMIN, S., **Simulated Jumping**, Annals of Operations Research 86, 23-38, 1999.

BODIN L.D. et al. **Routing and scheduling of vehicles and crews: The state of the art**. *Computers and Operations Research*, vol. 10, n.2, 1983.

BURIOL, Luciane Salete. **Algoritmo Memético para o Problema do Caixeiro Viajante Assimétrico como Parte de um Framework para Algoritmos Evolutivos**. Dissertação de Mestrado: UNICAMP, 2000.

CAMPELLO, R. Eduardo; MACULAN, Nelson. **Algoritmos e Heurísticas: Desenvolvimento e Avaliação de Performance**, Niterói, RJ, EDUFF, 1994.

CHIANG, W.-C.; RUSSELL, R. A. **Simulated annealing metaheuristics for the vehicle routing problem with time windows**. *Annals of Operations Research*, 63, 3-27, 1996.

CHIANG, W.-C; Russell, R. A. **A reactive tabu search metaheuristic for the vehicle routing problem with time windows**. *INFORMS Journal on Computing* 9, 417-430, 1997.

CHRISTOFIDES, N. **Travelling Salesman Problem**. Wiley Chichester, 1979.

COELHO, Leandro dos Santos. **Fundamentos, Potencialidades e Aplicações de Algoritmos Evolutivos**. - São Carlos, SP: SBMAC, 2003.

DAVIS, L. **Handbook of Genetic Algorithms**. Van Nostrand Reinhold. New York, 1991.

FEO, T. REZENDE, M., **A greed randomized adaptive search procedure for maximum independent set**, Operations Research 42, 860-879, 1994.

GARCIA, Vinícius Jacques. **Algoritmos meméticos paralelos aplicados a problemas de otimização combinatória**. Dissertação de Mestrado: UNICAMP, 2002.

GAREY, M. R.; JOHNSON, D. S., **Computers Intractability: A guide to the Theory of NP-Completeness**, Freeman, San Francisco, 1979.

GENDREAU, M.; HERTZ, A.; LAPORTE, G. **New Insertion and Post optimization Procedures for the Travelling Salesman Problem**. Operations Research 40, 1086-1094, 1992.

GEOFFRION, A. M.; GRAVES, G. W. **Multicommodity distribution system design by Benders decomposition**. *Management Science*, 20:822-844, 1974.

GLOVER, F. **Tabu Search: a Tutorial**, CAAI Report, University of Colorado, Boulder, 1990.

GLOVER, F., e LAGUNA, M., **Tabu Search**, Colin Reeves (ed.), em *Modern Heuristic Techniques*. Blackwell Scientific Publications, Oxford, Blackwell, 70-150, 1993.

GOLDBERG, D. E. **Genetic Algorithms in Search, Optimization, and Machine Learning**, Addison-Wesley, 1989.

GOLDBARG, M. C.; H. P. R. LUNA. **Otimização Combinatória e Programação Linear – Modelos e Algoritmos**. Rio de Janeiro: Editora Campus, 2000.

HOMBERGER, J.; GEHRING, H. **Two evolutionary metaheuristics for the vehicle routing problem with time windows**, *Information Systems and Operational Research – Special issue: Metaheuristics for location and routing problems*, 1999.

KIRKPATRICK, S., GELLAT, C.D. VECCHI, M.P., **Optimization by Simulated Annealing**, *Science* 220 (4598), 671 – 679, 1983.

LAPORTE G. et al. **Classical and modern heuristics for the vehicle routing problem**. *International Transactions in Operational Research*, v.7, n4/5, p.285-300, 2000.

LIN, S. **Computer solutions of the Traveling Salesman Problem**. *Bell System Computer Journal*, Vol. 44, pp. 713-733, 1965.

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

LIU, F.; SHEN, S. **A route-neighbourhood-based metaheuristic for vehicle routing problem with time windows**. *Working paper, National Chiao University, Hsinchu, Taiwan, 1998*.

MOSCATO, P. **On Evolution, Search, Optimization, Genetic Algorithms, and Matial Arts: Towards Memetic Algorithms**, Technical Report, Caltech Concurrent Computation Program, C3P Report 826, 1989.

MÜLLER, Felipe Martins. **Algoritmos Heurísticos e Exatos para Resolução do Problema de Seqüenciamento em Processadores Paralelos**. Tese de Doutorado, Faculdade de Engenharia Elétrica, Universidade Estadual de Campinas, Campinas, 1993.

NISSEN, V. **Evolutionäre Algorithmen**. *Deutscher Universitäts-Verlag, Wiesbaden*, 1994.

OSMAN, I. **Heuristics for Combinatorial Optimization Problems: Developments and New Directions**, In *proceedings of the first seminar on information technology and applications, Marfield Conference Centre, September, 1991*.

OSMAN, I. **Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem**. *Annals of Operations Research*, 41, 421-451, 1993.

PEREIRA, M. R. **Heurísticas e Metaheurísticas para o Problema do Caixeiro Viajante com Demandas Heterogêneas**. Monografia de Conclusão de Curso – Ciência da Computação, Universidade Federal de Santa Maria, 2004.

POTVIN, J. Y., **The traveling salesman library**, *ORSA Journal on Computing* 3, 376-384, 1993.

POTVIN, J. Y. **The Traveling Salesman Problem: A Neural Network Perspective**. *ORSA Journal on Computing* 5, 328-348, 1993.

POTVIN, J.-Y.; KERVAHUT, T.; GARCIA, B.-L.; ROUSSEAU, J.-M. **The Vehicle Routing Problem with Time Windows - Part I: Tabu search**. *INFORMS Journal on Computing*, search 8:158-164, 1996.

RECHENBERG, I. **Cybernetic Solution Path of an Experimental Problem**, Roy. Aircr. Establ. Libr. Transl., 1122, Farnborough, Hants, UK, 1965.

RECHENBERG, I. **Evolution strategie: Optimierung Technischer Systeme nach Prinzipien der Biologischen Evolution**. Frommann-Holzboog Verlag, 1973.

SARUBBI, J. F. M. **Um Modelo Linear para o Problema do Caixeiro Viajante com Demandas Heterogêneas**. Dissertação de Mestrado, Universidade Federal de Minas Gerais, 2003.

SCHWEFEL, H.-P. **Kybernetische Evolution als Strategie der Experimentellen Forschung in der Strömungstechnik**. *Diploma thesis, Technical University of Berlin*, March, 1965.

SOLOMON, M. M. **Algorithms for the vehicle routing and scheduling problems with time window constraints**. *Operations Research*, v. 35, n. 2, p. 254-265, 1987.

STARKWEATHER, T. et alii. **A comparison of genetic sequencing operators**. In: INTERNATIONAL CONFERENCE ON GENETIC ALGORITHMS, 1991. Los Altos, CA: Morgan Kaufmann, 1991. p. 69-76.

STÜTZLE, T.; M. DORIGO. **“ACO Algorithms for the Traveling Salesman Problem”**. To appear in K. Miettinen, M. Mäkelä, P. Neittaanmäki e J. Periaux, (eds), *Evolutionary Algorithms in Engineering and Computer Science: Recent Advances in*

Genetic Algorithms, Evolution Strategies, Evolutionary Programming, Genetic Programming and Industrial Applications, John Wiley & Sons, 1999.

TAILLARD et al. ***A tabu search heuristic for the vehicle routing problem with soft time windows.*** *Technical report CRT-95-66, Centre de recherche sur les transports, Université de Montréal, Montréal, 1996.*

THANGIAH et al. , S. R. ***GIDEON: a genetic algorithm system for vehicle routing with time windows.*** *In Proceedings of the 7th Conference on Artificial Intelligence for Applications, pp. 322-328. IEEE Press, Miami, FL, 1991.*