



**Business Intelligence**

**PUC**  
RIO

*Lúcio Rogério Botêlho*

*Alocação de Recursos nas Atividades de  
Intervenções em Poços de Petróleo Através  
de Algoritmos Genéticos*

*Monografia de Final de Curso*

*27/12/2011*

***Monografia apresentada ao Departamento de Engenharia Elétrica  
da PUC/Rio como parte dos requisitos para a obtenção do título  
de Especialização em Business Intelligence***

***Orientadores:***

Prof. Dr. Marco Aurélio C. Pacheco

Prof. Dr. André Vargas Abs da Cruz

---

## Dedicatória

Ao meu filho Pablo que foi gerado e nasceu durante o BI Master

## **Agradecimentos**

À Deus que sempre me encaminha

À minha esposa Lys e ao meu filho Pablo nascido durante o curso, por seu amor, carinho e por terem paciência e abdicarem de minha presença em determinados momentos devido às minhas tarefas deste desenvolvimento.

Aos meus pais, Lauro e Alcione, e minha irmã Thayenne, por seu amor e carinho, e que apesar de não estarem próximos sempre me auxiliam e estimulam.

Ao meu orientador Marco Aurélio Pacheco e ao meu co-orientador André Vargas Abs da Cruz por terem me guiado ao longo deste trabalho e me auxiliado em diversos questionamentos.

Ao Filipe Ferreira Mazzini e ao Mayron Rodrigues de Almeida colegas da Petrobras TIC/TIC-SERV/SPO (área de Pesquisa Operacional) que me apoiaram neste trabalho com informações relacionadas ao problema tratado neste texto.

Aos professores do curso de BI Master que me possibilitaram um acréscimo de novas informações e pelo convívio neste meio.

Aos meus colegas e amigos da turma de BI Master pelo companheirismo e compartilhamento de idéias, com um agradecimento especial ao pessoal da Petrobras Roberto, Renata, Fabrício, Kauê e Bruno.

Ao meu gerente Fernando da Silva Rodrigues e ao meu coordenador anterior Paulo Romero Zanconato Pinto e a atual Jacilene Silva Torres por terem possibilitado que eu realizasse este curso e projeto para o meu desenvolvimento tanto pessoal quanto profissional.

A todas as outras pessoas que me apoiaram ou ajudaram de uma forma ou de outra ao longo deste trabalho.

À Petrobras por ter financiado este estudo e ter possibilitado um incremento cultural e intelectual às minhas atividades na empresa.

## Resumo

As intervenções em poços de petróleo requerem um conjunto de recursos como materiais, equipamentos e serviços que devem estar disponíveis para uma sonda realizar as suas atividades de Perfuração, Completação entre outras. Estes recursos são críticos para que as sondas consigam desempenhar o seu trabalho no prazo definido, uma vez que o atraso em suas atividades acarreta o pagamento de valores exorbitantes de aluguel diário de sonda.

A distribuição destes recursos entre as sondas deve ser realizada de forma a minimizar os tempos de espera e com isso diminuir os gastos com sondas e equipamentos.

Existem algumas técnicas para se tratar este tipo de problema, sendo mais focadas em analisar todo o espaço de busca, contudo o tempo gasto para esta análise cresce exponencialmente de acordo com a quantidade de recursos por atividades. Estes recursos vêm sendo cada vez mais procurados e sendo necessário aumentar a sua quantidade ao longo dos anos. Desta forma esta monografia visa criar um protótipo utilizando a meta-heurística de Algoritmos Genéticos para tentar resolver este problema em uma fração de tempo de algumas outras soluções, sendo menos dependente da quantidade de recursos, pois de acordo com sua natureza probabilística ele não deve percorrer todo espaço de busca.

## **Abstract**

The interventions in oil wells require several resources such as materials, equipment and services that should be available to a drilling rig conduct its Drilling, Completion and others activities. These resources are critical to the rigs be able to carry out their work in scheduled time, since the delay in their activities charge the payment of exorbitant daily rental availability costs.

The distribution of resources between rigs must be done to minimize waiting times, thus reducing money spent on equipment and rigs rental.

There are some techniques to handle this kind of problem, and these techniques are mainly focused in examination of the entire search space. However the time taken for this analysis grows exponentially with the amount of resources per activities.

These resources have been increasingly requested, and its amount must have been expanded over the years. Thus this work aims to create a system prototype using Genetic Algorithms meta-heuristics to deal with this problem and maybe find a solution in a fraction of the time spent by other solutions. Genetic Algorithms are less dependent on the amount of resources, since according to their probabilistic nature it should not seek the entire search space.

## Sumário

|        |                                      |    |
|--------|--------------------------------------|----|
| 1      | Introdução.....                      | 7  |
| 1.1.   | Motivação.....                       | 8  |
| 1.2.   | Objetivos do Trabalho .....          | 8  |
| 1.3.   | Descrição do Trabalho .....          | 9  |
| 1.4.   | Organização da Monografia .....      | 10 |
| 2      | Descrição do Problema .....          | 11 |
| 3      | Metodologias .....                   | 16 |
| 3.1.   | Avaliação .....                      | 17 |
| 3.2.   | Seleção de Indivíduos .....          | 18 |
| 3.3.   | Cromossomo.....                      | 18 |
| 3.4.   | Operadores Genéticos .....           | 18 |
| 3.4.1. | Operador Genético de Crossover.....  | 18 |
| 3.4.2. | Operador Genético de Mutação .....   | 19 |
| 3.5.   | Elitismo .....                       | 20 |
| 4      | Arquitetura do Sistema Proposto..... | 21 |
| 4.1.   | Representação do Cromossomo.....     | 22 |
| 4.2.   | Processo de Crossover .....          | 24 |
| 4.3.   | Processo de Mutação.....             | 26 |
| 4.4.   | Avaliação da População.....          | 28 |
| 4.4.1. | Penalização de Indivíduos.....       | 28 |
| 4.4.2. | Avaliação do Custo .....             | 29 |
| 5      | Resultados.....                      | 31 |
| 5.1.   | Execução de Testes.....              | 31 |
| 5.2.   | Dados Utilizados .....               | 33 |
| 6      | Conclusões e Trabalhos Futuros.....  | 35 |
| 6.1.   | Trabalhos Futuros.....               | 36 |
|        | Referências Bibliográficas.....      | 37 |

## 1 Introdução

Com o crescente número de projetos para desenvolvimento de campos de petróleo tanto existentes quanto os descobertos recentemente com a iniciativa do pré-sal é cada vez mais necessária a utilização de sondas (figura 1) para prospectar estes campos realizando os trabalhos nos inúmeros poços de petróleo de cada campo.

Os trabalhos realizados pelas sondas de perfuração, sondas de completação entre outras, nos poços são chamados de intervenções em poços, e são classificados de acordo com os conceitos e convenções do programa Pró-Poço da Petrobras em quatro tipos de intervenções: Perfuração, Completção, *Workover* (Retrabalho) e Avaliação Exploratória.

Para cada um destes tipos de intervenções existe um conjunto de atividades que podem ser realizadas pelas sondas de perfuração em conjunto com outros equipamentos. As sondas de perfuração são recursos muito caros, chegando a centenas de milhares de dólares o custo de aluguel diário, tendo como tempo mínimo de contratação do serviço o período de um ano (Mazzini et al, 2010).

Para o funcionamento da sonda, de acordo com a atividade a ser realizada, um conjunto de equipamentos diferente é necessário, sendo que diversos equipamentos são compartilhados entre as sondas. Ou seja, é um tipo de recurso crítico, que apesar de individualmente seu valor ser ínfimo em relação ao valor diário de sonda, o conjunto total de equipamentos contratados ao longo de um ano pode superar o valor de diárias de sonda. Além disso, também deve ser levado em consideração o tempo de deslocamento de sonda entre os poços e ainda entre campos de diferentes regiões que acarretam um maior atraso.

A indisponibilidade de um único recurso crítico de equipamento pode acarretar na parada da sonda de perfuração, e com isso o pagamento de diárias com a sonda ociosa, além do que também afetaria o cronograma de outras sondas. E se for analisado a longo prazo este atraso irá gerar também um atraso do início da entrada em produção de um ou mais poços dependentes desta sonda.

## 1.1. Motivação

A sonda de perfuração, por seu valor individual, é considerado um recurso crítico separado dos demais equipamentos para a Petrobras, e a sua alocação é tratada de forma independente por sistemas de cronograma de sondas (Lazzarine et al, 2008) e pelas equipes gerenciais que usualmente utilizam as informações destes sistemas para suas tomadas de decisão.

Após o cronograma de sondas ter sido definido para alocar sondas em diversos campos, é necessário tratar o problema de alocação de recursos críticos às atividades de intervenções em poços nestas sondas.

Uma forma de resolver este problema é através de Programação Matemática (Mazzini et al, 2010) que percorre todo o espaço de busca, contudo à medida que aumenta a quantidade de sondas e também a quantidade de recursos por atividade, fica mais difícil de ser resolvido desta forma uma vez que leva mais tempo já que é um problema de ordem de grandeza exponencial.

Este trabalho visa resolver este problema utilizando meta-heurística de Algoritmos Genéticos que mesmo não percorrendo todo o espaço de busca, consegue obter soluções otimizadas, contudo não garantindo que sejam sempre soluções ótimas.

## 1.2. Objetivos do Trabalho

Os objetivos deste trabalho são:

- O desenvolvimento de um protótipo de sistema para alocação de recursos utilizando Algoritmos Genéticos;
- A criação de um método escalável para alocação de recursos em atividades de sondas de perfuração;
- A definição de uma forma otimizada de estimar a quantidade de recursos para as atividades ao longo de um período de tempo;



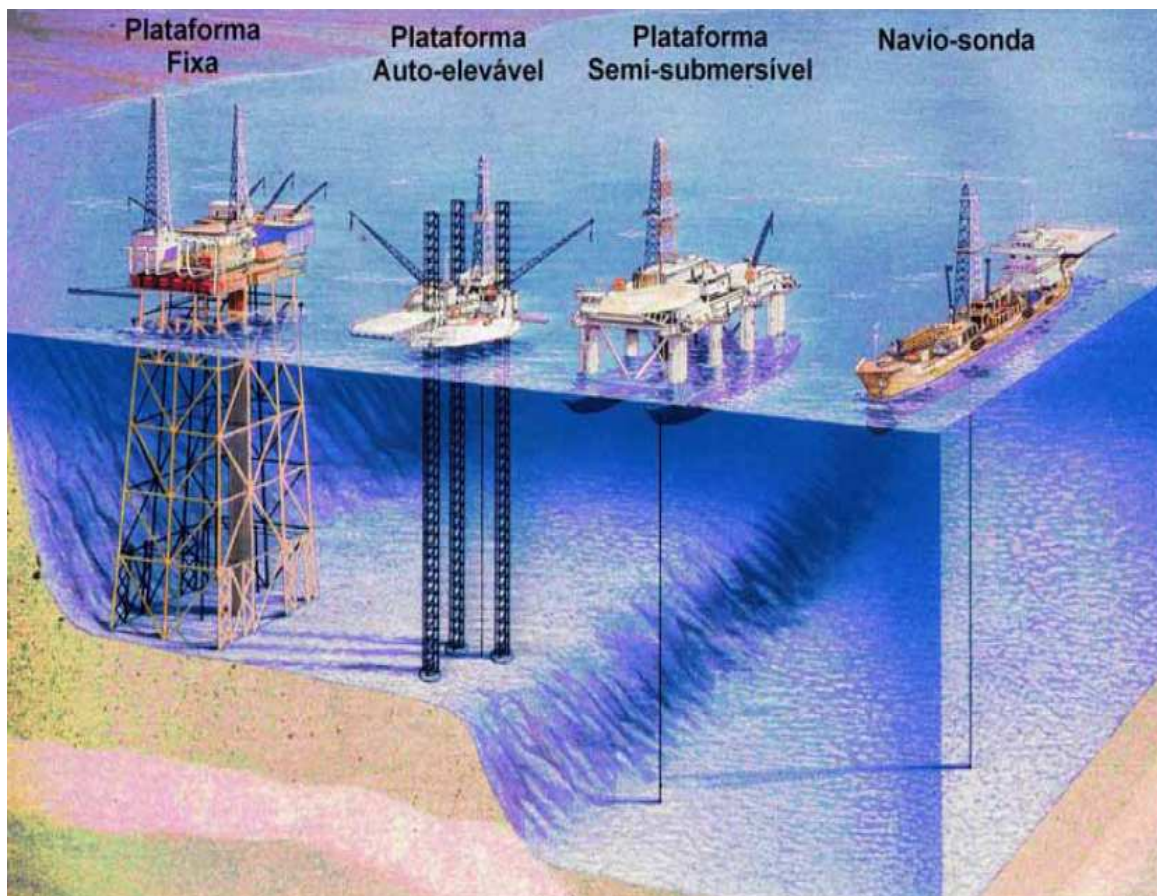


Figura 1 – Sondas instaladas em plataformas

### 1.3. Descrição do Trabalho

O desenvolvimento dessa monografia envolveu algumas etapas: Estudo de características e métodos empregados em Algoritmos Genéticos; Estudo de bibliotecas de Algoritmos Genéticos em Java; Desenvolvimento de um protótipo com simplificações em relação ao problema inicial; Montagem de uma base para testes; Utilização de um Estudo de Caso real.

O estudo de características e métodos empregados em Algoritmos Genéticos serviu para definir quais os tipos de representação de cromossomos, crossover, mutação, seleção de população, testados e utilizados ao longo do desenvolvimento.

O estudo das bibliotecas de Algoritmos Genéticos em Java serviu para a escolha da biblioteca mais robusta, prática, estável e com um amplo leque de implementações de características interessantes ao problema a ser resolvido.

O desenvolvimento de um protótipo se fez necessário para se colocar em prática os conhecimentos adquiridos ao longo do BI Master, bem como para testar na prática a teoria apresentada neste trabalho.

Foi criada inicialmente uma base de dados para testes de forma a testar as versões iniciais do protótipo e aperfeiçoá-lo para ser utilizado por um estudo de caso real. Finalmente, o estudo de caso real serviu para verificar a robustez e qualidade da solução em comparação com outras formas de se resolver este problema.

#### **1.4. Organização da Monografia**

Esta monografia está dividida em cinco capítulos adicionais, descritos a seguir:

O capítulo 2 apresenta a descrição do problema de alocação de recursos críticos nas atividades de intervenções de poços de petróleo.

O capítulo 3 descreve as metodologias escolhidas para solucionar o problema em questão e também compara com outras metodologias presentes na literatura.

O capítulo 4 mostra detalhes da arquitetura utilizada na criação do protótipo do sistema.

O capítulo 5 detalha os resultados obtidos com as pesquisas e desenvolvimentos realizados durante este trabalho.

Finalmente, o capítulo 6 descreve as conclusões deste trabalho e identifica possíveis trabalhos futuros.

## 2 Descrição do Problema

As intervenções em poços de petróleo requerem um conjunto de recursos como materiais, equipamentos e serviços que devem estar disponíveis para realizar as suas atividades necessárias. Para cada tipo de intervenção existem várias atividades que podem ser realizadas pelas sondas e um determinado conjunto de recursos (Figura 2).

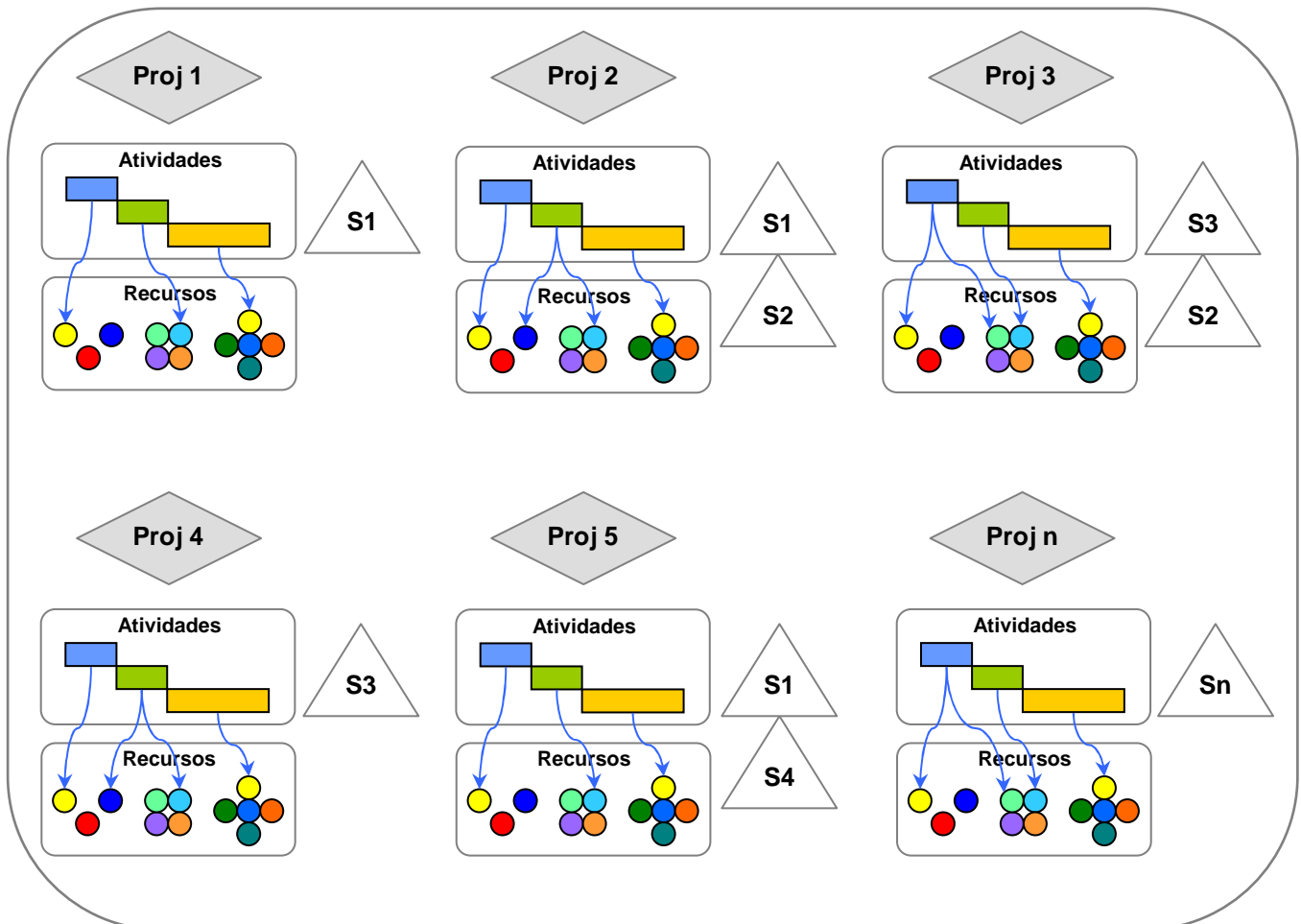


Figura 2 – Projetos com suas sondas, realizando atividades com recursos alocados

As sondas de perfuração são recursos muito caros, chegando a centenas de milhares de dólares o custo do aluguel diário, e tendo como tempo mínimo de contratação do serviço o período de um ano (Mazzini et al, 2010). Estas sondas possuem uma grande demanda entre as empresas de prospecção de petróleo, sendo bastante escassas.

Segundo Mazzini et al (2010) cada sonda de perfuração, por seu valor individual, é considerada um recurso crítico separado dos demais recursos para a

Petrobras, e a sua alocação é tratada de forma independente por sistemas de cronograma de sondas (Lazzarine et al, 2008) e pelas equipes gerenciais que usualmente utilizam as informações destes sistemas para suas tomadas de decisão.

O texto de Mazzini et al (2010) mostra que o dimensionamento de sondas e o sequenciamento de projetos de perfuração e exploração com alocação de sondas são dois problemas conhecidos. Sendo que o problema de dimensionamento de sondas freqüentemente é tratado como um problema de simulação (Shyshou, 2010). Já o problema de sequenciamento de projetos com restrição de recursos é visto como um problema de programação matemática.

Alguns autores, (Blazewicz et al, 1991) e (Mazzini, 2007) apresentaram representações de programação inteira para este problema, mas por se tratar de um problema de complexidade não polinomial existem implementações utilizando heurísticas e meta heurísticas como Algoritmos Genéticos (Vasconcellos et al, 2006) e Busca Tabu que podem ser mais escaláveis para esta categoria de problemas.

Após o cronograma de sondas ter sido definido para alocar sondas em diversos campos, é necessário tratar o problema de alocação de recursos críticos às atividades de intervenções em poços nestas sondas.

Neste problema, dos recursos descritos acima, levamos somente em consideração os equipamentos, pois estes são contratados por períodos mínimos de um ano ou mais. Devem ser contratados com bastante antecedência, chegando a um ano antes de sua disponibilidade. Estes contratos cobram tanto pela disponibilidade do equipamento quanto por sua utilização.

Diferentemente dos equipamentos, os materiais de consumo não são considerados como variáveis neste problema, pois são utilizados uma única vez e são adquiridos com facilidade e não contratados.

Os serviços também não são considerados como variáveis deste problema.

A alocação de equipamentos críticos visa atender as atividades previstas no cronograma de sondas. O cronograma de sondas reúne diversos projetos de avaliação exploratória de campos e poços (que é uma fase onde está se buscando trabalhar novos poços em regiões não exploradas ou pouco exploradas) e desenvolvimento de poços da Petrobrás (tratar regiões já conhecidas explorando poços existentes ou novos com grande possibilidade de se encontrar hidrocarbonetos).

Para o funcionamento da sonda, de acordo com a atividade a ser realizada, um conjunto de equipamentos diferente necessita estar disponível para o início desta atividade. Sendo que diversos equipamentos são compartilhados entre as várias sondas de projetos simultâneos. Ou seja, um equipamento é um tipo de recurso crítico, que apesar de individualmente seu valor ser ínfimo em relação ao valor diário de sonda, o conjunto total de equipamentos contratados ao longo de um ano pode superar o valor de algumas diárias de sonda.

A distribuição destes equipamentos entre as sondas deve ser realizada de forma a minimizar os tempos de espera e com isso diminuir os gastos com sondas e também tentar diminuir os custos de contratação de equipamentos.

Outro fator levado em consideração é o do sub-dimensionamento de equipamentos gerado pela redução de custos na contratação destes equipamentos que pode levar a atrasos no cronograma.

Estes recursos são críticos para que as sondas consigam desempenhar o seu trabalho no prazo definido, uma vez que o atraso em suas atividades acarreta o pagamento de valores exorbitantes de aluguel diário de sonda. As atividades atrasadas podem também gerar atrasos em outras atividades dependentes, bem como gerar atrasos em outras sondas relacionadas e assim progressivamente.

Além disso, também deve ser levado em consideração o tempo de deslocamento de sonda, e dos recursos, entre os poços e ainda entre campos de diferentes regiões que acarretam um maior atraso.

Neste problema não é permitido alterar a seqüência dos projetos no cronograma realizado pelo processo anterior, contudo podemos atrasar atividades que não consigam obter os recursos necessários à sua execução.

Desta forma, a alocação de equipamentos críticos deve procurar atender as demandas por recursos das atividades, visando à minimização do custo de contratação de equipamentos somado ao custo das diárias de sondas paradas.

Descrição das principais regras e definições do problema em questão:

1. O cronograma de sondas é um dado de entrada, contendo os relacionamentos entre os projetos, sondas, atividades, demandas de atividade por recursos e finalmente recursos. Abaixo está uma lista dos meta-dados dos principais dados informados pelo cronograma de sondas disponíveis para este trabalho de alocação de recursos:

- a. Projeto: código, nome, início, término;
  - b. Sonda: código, nome, valor da diária;
  - c. Atividade: código, nome, início, duração, código da sonda, código do projeto;
  - d. Precedência: código atividade, código da atividade sucessora, tipo de precedência;
  - e. Recurso: código, nome, valor da diária;
  - f. Demanda de Atividade por Recurso: código da atividade, código do recurso, quantidade deste tipo de recurso, início da demanda, duração da demanda;
2. Cada atividade possui uma lista de demandas por recursos, que devem ser atendidas em determinado prazo para não acarretar atrasos a esta atividade;
  3. Os relacionamentos de dependências (tipo de precedência) entre as atividades podem ser da forma Início-a-Início (II), Início-a-Término, Término-a-Início e Término-a-Término (TT);
  4. Uma atividade pode somente ser atrasada, não pode ser adiantada, devido as restrições impostas pelo cronograma de sonda para estes recursos;
  5. As atividades são atrasadas quando não há disponibilidade de algum dos equipamentos requisitados para atender uma de suas demandas;
  6. O atraso de uma atividade se propaga para uma atividade sucessora dependente;
  7. O atraso de uma ou mais atividades pode atrasar um projeto inteiro;
  8. O atraso de um projeto pode atrasar outro projeto que esteja na dependência de uma sonda ou outro recurso utilizado pelo primeiro;
  9. O custo do atraso das atividades é medido pelo custo da diária de sonda;
  10. Os equipamentos são contratados por períodos mínimos, sendo maiores ou iguais a um ano;
  11. O custo de contratação de um equipamento é dado pelo seu custo de disponibilidade mais seu custo de operação. Ou seja, se um equipamento está em uso é cobrado o custo de operação que é maior que o custo de disponibilidade. Sendo que o custo de disponibilidade é

cobrado quando o equipamento está ocioso, mas está disponível para a utilização pela empresa.

Para se ter uma idéia do quantitativo de dados para o ano de 2010, o conjunto de todas as atividades está em torno de 2000 registros, o conjunto de demandas de atividade por recursos está em torno de 5000 registros, e finalmente o conjunto de recursos está em torno de 60 tipos diferentes de recursos. Por serem dados sigilosos o quantitativo de projetos bem como o quantitativo de sondas não serão registrados neste trabalho.

### 3 Metodologias

Os Algoritmos Genéticos tiveram seu desenvolvimento iniciado por John Holland (Holland, 1992), um tipo de algoritmo baseado nas teorias evolucionistas de Charles Darwin, Alfred Russel Wallace e Gregor Mendel, que descobriram que a mutação e a recombinação genética em uma população de indivíduos sexuados geram uma variabilidade entre estes indivíduos.

Este tipo de algoritmo (Haupt et al, 2004) parte de algumas possíveis soluções escolhidas aleatoriamente baseadas no conjunto de possíveis soluções (Espaço de Busca) para gerar a sua população inicial de indivíduos (representados por cromossomos). Esta população inicial é avaliada de acordo com os critérios de avaliação do problema. De acordo com estes critérios de avaliação é definido se a busca pela solução continua. Contudo, na maioria das vezes a busca por uma solução termina quando a evolução atingiu um determinado número de gerações.

Para o caso de continuação, alguns indivíduos da população atual são escolhidos para que sejam reproduzidos por cruzamento (*crossover*) ou para que seja realizada mutação e com isso criar uma nova população, verificando se as possíveis soluções estão melhorando em relação às gerações anteriores. Este processo vai sendo repetido até que se encontre um critério de término.

Alguns indivíduos vão sendo escolhidos para compor a população seguinte, podendo ser escolhidos indivíduos tanto mais promissores quanto menos promissores. Contudo, os indivíduos mais promissores geralmente têm mais chances de serem selecionados de forma a ir melhorando as gerações seguintes ("Quanto melhor um indivíduo se adaptar ao seu meio ambiente, maior será sua chance de sobreviver e gerar descendentes").

Os algoritmos genéticos são métodos de busca global que, diferentemente dos algoritmos tradicionais, não percorrerem todo o Espaço de Busca. Os métodos de busca tradicionais realizam uma busca exaustiva à procura da melhor solução, porém dependendo do tipo de problema a ser resolvido, esta forma pode ser inviável para um grande número de combinações de possíveis soluções (Michalewicz, 1996), levando a um processamento muito longo que duraria dias, meses, anos ou ainda não ter fim. Como os algoritmos genéticos não percorrem totalmente o Espaço de Busca, pois vão mudando sua escolha de acordo com a



evolução é possível encontrar soluções em tempos viáveis. Contudo estes algoritmos evolutivos não garantem que seja encontrada a melhor solução (ótima), mas no geral podem encontrar soluções sub-ótimas que para determinados tipos de problema já é suficiente.

Exemplo de algoritmo genético básico, baseado em (Michalewicz, 1996):

```

procedimento algoritmo genético
início
  t ← 0
  inicializar população P(t)
  avaliar P(t)
  enquanto (não condição de término) faça
  início
    t ← t + 1
    selecionar P(t) a partir de P(t - 1)
    crossover P(t)
    mutação P(t)
    avaliar P(t)
  fim
fim

```

### 3.1. Avaliação

Ao realizarmos uma busca no espaço de busca, onde são consideradas todas as possíveis soluções para o problema a ser resolvido, deve haver uma função de avaliação (ou função de custo) que seja capaz de avaliar os elementos do espaço de busca.

Ao longo das gerações do algoritmo genético ou sua evolução, cada indivíduo da população atual é avaliado recebendo uma nota, ou índice de aptidão que reflete sua capacidade de adaptar-se ao ambiente em que se encontra. Destes indivíduos os mais aptos tendem a se manter na população seguinte, enquanto os menos aptos tendem a desaparecer.

Desta forma os algoritmos genéticos direcionam sua busca sempre para regiões de mais alta aptidão, ou seja, para as regiões onde a função a ser minimizada (ou maximizada) possui valores mais baixos (ou mais altos).

### **3.2. Seleção de Indivíduos**

A seleção determina quais indivíduos da população serão escolhidos para se reproduzirem ou realizar mutação. Existem vários métodos de seleção (Michalewicz, 1996; Rothlauf, 2006), e alguns deles dão preferência aos indivíduos com maior índice de aptidão.

### **3.3. Cromossomo**

O cromossomo representa um conjunto de dados de uma possível solução para o problema sendo tratado. Um conjunto de genes forma o cromossomo, e dependendo de cada problema a ser tratado um tipo diferente de representação é utilizada como por exemplo representação binária (11100110101011), representação por números reais (9.4, 12.8, 7.2, 4.5, 1.1), representação por ordem (D, A, C, B, F, E), etc (Rothlauf, 2006).

### **3.4. Operadores Genéticos**

Os operadores genéticos são utilizados com o intuito de modificar a população de cromossomos ao longo das sucessivas gerações, de forma a ir melhorando os indivíduos desta população até encontrar a solução desejada ou ainda um resultado satisfatório. Estes operadores são úteis para a população se diversificar e dar continuidade às características dos melhores indivíduos nas próximas gerações (Rothlauf, 2006).

#### **3.4.1. Operador Genético de Crossover**

O operador de Crossover ou Cruzamento atua sobre alguns indivíduos da população previamente selecionados e realiza uma recombinação entre os seus genes durante esta reprodução, de forma que características são trocadas e levadas para os seus descendentes.

Caso não haja cruzamento, os descendentes serão cópias dos seus pais. Estes cruzamentos são uma tentativa de geração de indivíduos que contenham boas características de ambos os pais, e caso sejam melhores a tendência de acordo com a seleção é que se mantenham na população e sejam perpetuados.

Existe uma probabilidade chamada de taxa de crossover que determina a frequência de cruzamento dos indivíduos.

|                |                         |
|----------------|-------------------------|
| <b>Pai 1</b>   | <b>1000011010111101</b> |
| <b>Pai 2</b>   | <b>0110101101100100</b> |
|                |                         |
| <b>Filho 1</b> | <b>100001101100100</b>  |
| <b>Filho 2</b> | <b>0110101010111101</b> |

Tabela 1

A Tabela 1 exibe um exemplo de crossover utilizando cromossomos com representação binária. Um ponto de crossover é escolhido aleatoriamente em dois indivíduos de modo que a recombinação que irá gerar seus filhos fique com a primeira parte de um indivíduo e a segunda parte do outro indivíduo e vice-versa.

Este exemplo ilustrou o que é um crossover de um ponto, contudo podem ser utilizados crossover de dois ou mais pontos, ou ainda o crossover uniforme que pode utilizar um padrão ou máscara para realizar a recombinação.

A representação utilizada no cromossomo afetará diretamente a forma como um operador deste tipo é criado.

### 3.4.2. Operador Genético de Mutação

Este operador é fundamental para desenvolver a diversidade genética da população, introduzindo novas características ou modificando às existentes. Esta diversidade atua levando à exploração de outras regiões no espaço de busca, evitando que o algoritmo se estabilize em um ponto de mínimo ou máximo local, por exemplo.

De acordo com uma probabilidade chamada de taxa de mutação, cada indivíduo da população pode sofrer algum tipo de mutação em um ou mais genes. A taxa de mutação geralmente é utilizada com valores pequenos podendo ser

aumentada nas gerações finais para melhorar a pressão de seleção e evitar a convergência prematura. Caso a taxa de mutação seja 1 então todos os cromossomos serão alterados, e se for 0 nenhum deles sofrerá alteração. Caso a mutação ocorra com excessiva frequência o algoritmo genético se tornará uma Busca Aleatória.

A seguir, tabela 2, é exibido um par de exemplos usando cromossomos com representação binária, onde alguns bits são alterados de 1 para 0 ou vice-versa.

|                |                         |
|----------------|-------------------------|
| <b>Pai 1</b>   | <b>1010110011000111</b> |
| <b>Filho 1</b> | <b>1010010011000111</b> |
|                |                         |
| <b>Pai 2</b>   | <b>1111000110110101</b> |
| <b>Filho 2</b> | <b>1101000110111101</b> |

Tabela 2

Como existe uma variedade de representações de cromossomos, a forma como é aplicada a mutação dependerá diretamente de cada uma destas representações.

### 3.5. Elitismo

Ao gerar uma nova população utilizando crossover e mutação pode acontecer de perdermos os melhores indivíduos devido a estas alterações para gerar a população seguinte. Contudo podemos copiar para a população seguinte os melhores indivíduos da população atual, de forma que não sejam perdidos, este método é chamado de Elitismo. Os indivíduos restantes para a população seguinte serão gerados usando crossover e mutação, utilizando até os melhores indivíduos originais da população atual. Assim podemos evitar que as melhores soluções encontradas até o momento sejam apagadas.

## 4 Arquitetura do Sistema Proposto

Este problema de alocação de recursos em atividades com precedências, como pôde ser analisado no capítulo 2, não é um problema de solução muito fácil e nem mesmo rápida (Wall, 1996; Cruz, 2003). Devido às limitações de tempo de entrega deste trabalho foi desenvolvida uma arquitetura e também um protótipo de sistema para resolução do problema em questão, porém realizando algumas simplificações para se adequar ao tempo disponível. Esta arquitetura foi desenvolvida tendo em foco a possibilidade de extensão para o problema completo em um momento futuro.

Na implementação do protótipo foi utilizada a linguagem Java, pela sua grande difusão atual tanto no meio acadêmico quanto no meio empresarial, sem contar que é multiplataforma, flexível, escalável entre outros benefícios.

Após ter definido a linguagem de programação foi necessário analisar algumas bibliotecas ou frameworks de Algoritmos Genéticos em Java (ECJ, 2011; EvoJ, 2011; Jenes, 2011; JAGA, 2011; JGAP, 2011; WatchMaker, 2011). Tendo sido escolhido o JGAP (JGAP, 2011) que é um componente de Algoritmos Genéticos e Programação Genética na forma de um framework, que provê os mecanismos genéticos básicos a serem usados ou estendidos para resolução de diversos problemas. Esta ferramenta foi escolhida devido a possuir diversos operadores genéticos, possibilidade de extensão, possuir vários exemplos, código fonte aberto e de estar sendo utilizada por vários projetos acadêmicos ou comerciais e tendo aparecido em artigos de vários congressos (Ipate, 2008; Lefticaru, 2007; Lefticaru, 2008; Romera, 2010)

O conjunto de projetos, sondas, atividades e recursos, que foram descritos no capítulo 2 e também na figura 2, exibem um panorama geral do problema de alocação de recursos nas atividades com suas precedências. Sendo que estas atividades serão realizadas por projetos e necessitam de sondas.

Como já foi dito o problema de alocação de sondas (cronograma de sondas) já é resolvido por outro sistema servindo de subsidio para este protótipo, logo não é nosso foco representar diretamente nem os projetos e nem as sondas, uma vez que servem de coadjuvantes. Serão representadas diretamente as atividades com suas precedências e demandas, e os recursos a serem alocados.

#### 4.1. Representação do Cromossomo

Primeiramente foi necessário analisar uma forma de serem representadas as atividades com suas precedências, sendo talvez a mais prática e direta a representação por grafos, de forma que foram representadas estas associações usando um grafo direcionado ou orientado (Netto, 2006).

Cada nó ou vértice do Grafo de Atividades, figura 3, representa uma única atividade ( $A_i$ ), a sonda que está associada ( $S_j$ ) e uma lista de demandas por recurso ( $R_k$ ) daquela atividade, onde para cada uma destas demandas está registrada a data de início e a duração da demanda e a quantidade de recursos para que seja atendida. Os relacionamentos de dependências entre as atividades da forma Início-a-Início (II), Início-a-Término, Término-a-Início e Término-a-Término (TT) não serão contemplados inicialmente neste trabalho, somente as dependências simples do tipo Término-a-Início entre as atividades serão contempladas nesta fase do protótipo.

Para auxiliar na tarefa de representação, visualização e busca do Grafo de Atividades foi utilizado o framework JUNG - Java Universal Network/Graph Framework (JUNG, 2011), que é uma biblioteca que fornece uma linguagem comum e extensível para modelar, analisar e visualizar dados que possam ser representados como um grafo ou uma rede.

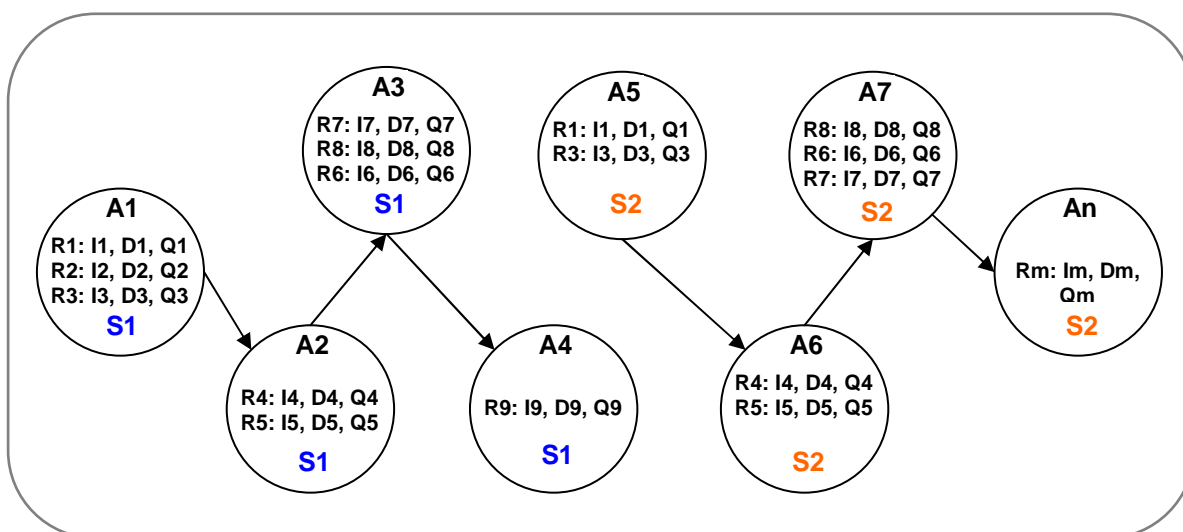


Figura 3 – Grafo representando dependências entre atividades (A1, A2, etc) das sondas S1 e S2

A representação do cromossomo depende diretamente deste grafo para gerar soluções viáveis. Contudo representar esta estrutura diretamente no cromossomo para cada indivíduo da população poderia ser computacionalmente muito custoso em termos de memória, de forma que poderíamos usar um mecanismo mais simples para esta representação. O mecanismo utilizado foi uma implementação do padrão de projeto *Proxy* (Gamma, 1995) que provê um substituto a um objeto de modo a controlar o acesso a este. Este padrão é aplicável quando existe a necessidade de uma referência a um objeto, porém mais versátil ou sofisticada que simplesmente um ponteiro (Gamma, 1995).

Baseado neste Grafo de Atividades, no *proxy* de atividade, e nos recursos a serem alocados foi projetado um cromossomo que possui dois segmentos, ou duas representações de conjuntos de genes (figura 4).

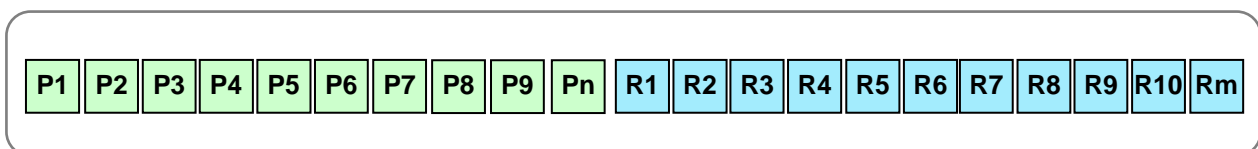


Figura 4 – Representação básica do cromossomo

O primeiro segmento representa o conjunto de genes que são *proxies* de atividades, ou seja, cada gene é um *proxy* para uma atividade no Grafo de Atividades (figura 5). Um *proxy* é representado neste exemplo como  $P_y$ , e um recurso por  $R_z$ .

Um *proxy*, além de apontar para sua atividade correspondente, permite acesso aos elementos desta atividade através de seus próprios métodos, bem como possui um atributo que informa a quantidade de demandas daquela atividade que já foram atendidas, para evitar buscas desnecessárias.

O segundo segmento representa o conjunto de genes que são recursos. Cada gene representa um tipo de recurso, e como vários recursos de um determinado tipo podem ser requisitados no mesmo período por uma ou mais atividades é necessário representar em cada gene uma lista contendo itens de recurso deste tipo a serem utilizados em paralelo. E cada item destes pode estar disponível para atender uma demanda de atividade sempre que não estiver sendo utilizado por outra, ou seja, este item de recurso pode servir a diversas atividades com períodos

de demanda não coincidentes, logo cada item de recurso possui uma lista para as atividades.

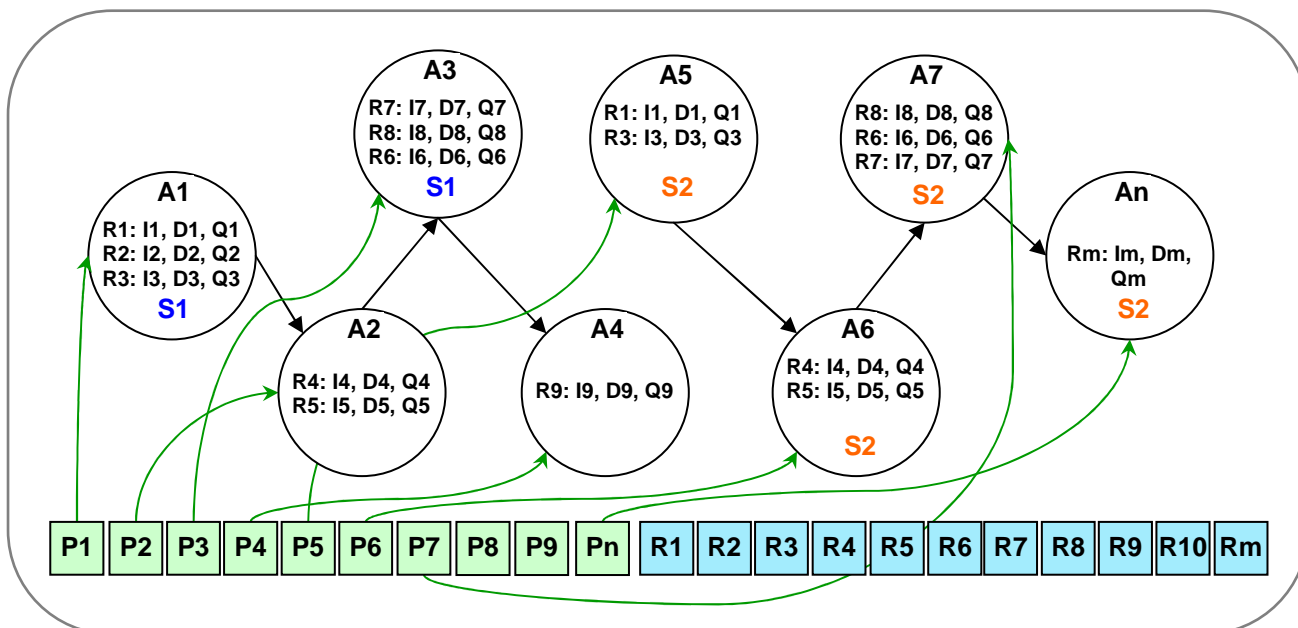


Figura 5 – Representação de genes *proxies* apontando para suas respectivas Atividades

## 4.2. Processo de Crossover

Uma vez que esta representação de cromossomo possui dependências entre os genes de recurso de um segmento com os genes de *proxy* de atividade do outro, o crossover tem que ser realizado de forma que não crie inconsistências nos novos cromossomos gerados.

O crossover é realizado segundo alguns critérios para sempre gerar indivíduos válidos. Primeiramente o crossover não age diretamente sobre o segmento de genes de *proxy* de atividade, ele atua nesta parte do cromossomo somente de forma secundária e totalmente dependente do que acontecer com o segmento de genes de recurso. Assim os pontos de crossover somente abrangerão os genes de recursos, e não os genes de *proxy*.

Tendo sido escolhido o ponto ou pontos de crossover no segmento de recursos dos cromossomos selecionados (figura 6), os genes de cada uma dessas seções são trocados de forma que um novo cromossomo fique com a primeira parte de um indivíduo e a segunda parte do outro indivíduo e vice-versa e assim sucessivamente de acordo com o número de seções (figura 7), como descrito para



o crossover binário no capítulo 3. As figuras 6 e 7 apresentam os itens de recurso  $I_x$  para cada tipo de recurso  $R_z$ .

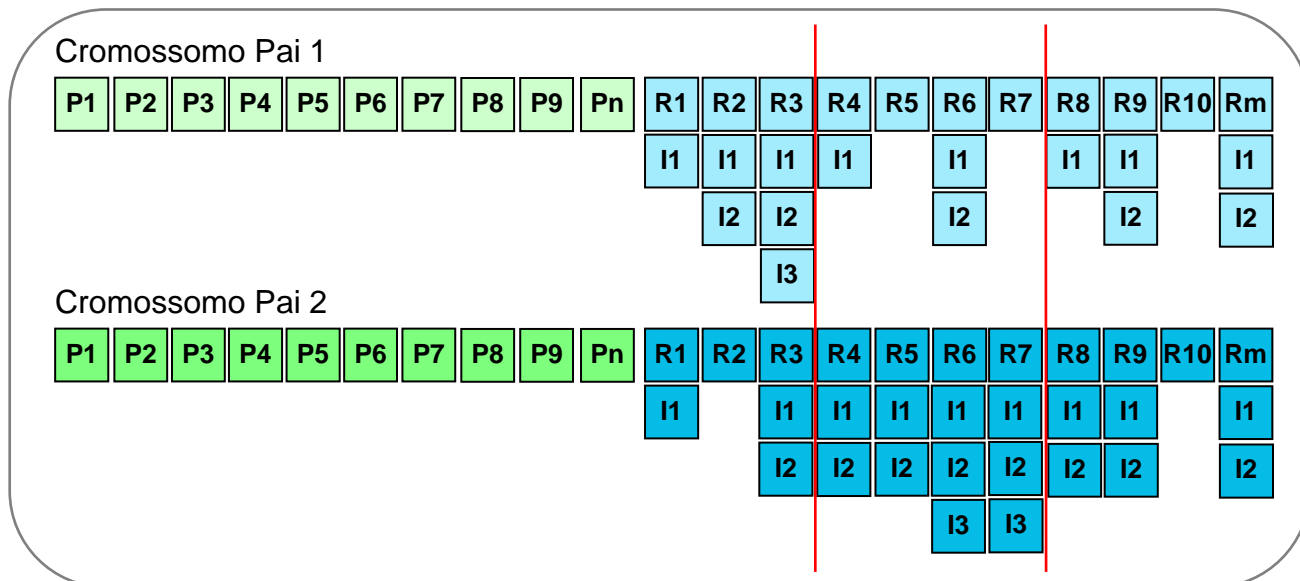


Figura 6 – Cromossomos com pontos de crossover definidos

Ao trocar os genes principais dos cromossomos a primeira parte da operação de crossover está concluída. Porém ainda é necessário que o segmento de *proxy* de atividade seja atualizado com estas novas informações trocadas.

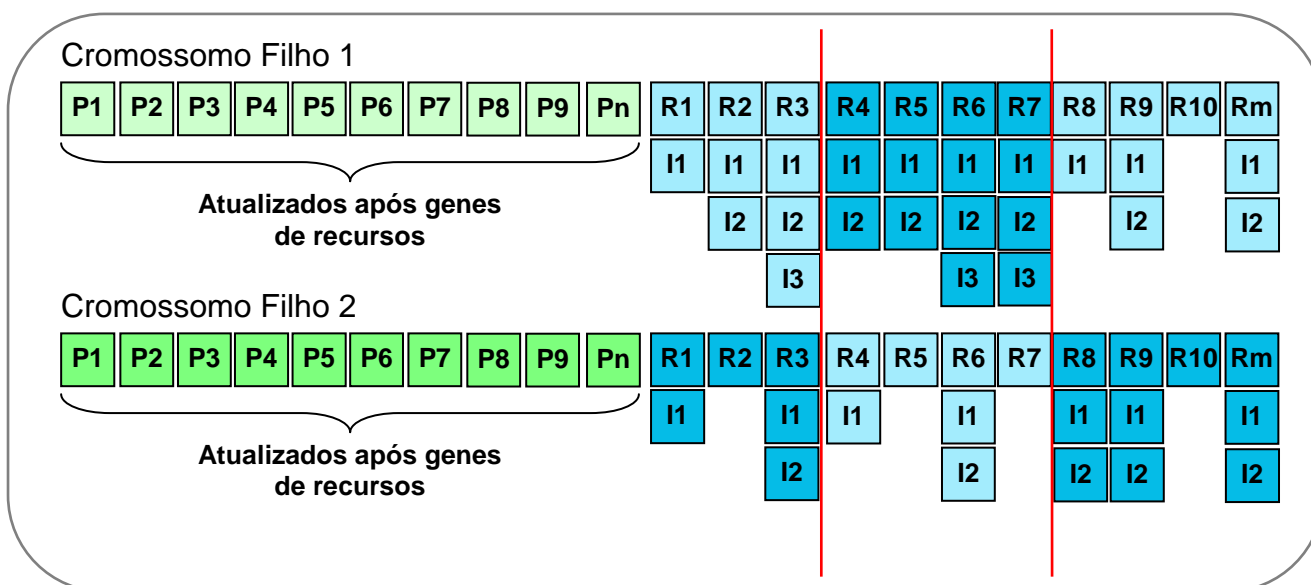


Figura 7 – Cromossomos gerados no crossover

Inicia-se um processo de varredura de todos os genes do segmento de *proxy* de atividade. Onde para cada *proxy* é feita uma verificação no segmento de

recursos pelos recursos que sua atividade está demandando e verifica se toda a demanda da sua atividade já foi atendida ou não, e finalmente é atualizado este gene de *proxy*.

### 4.3. Processo de Mutação

Novamente, como a representação de cromossomo possui dependências entre os genes de recurso de um segmento com os genes de *proxy* de atividades do outro, somente devem ser gerados novos cromossomos com de forma consistente.

O processo de mutação é realizado em cromossomos que passem por uma seleção previa. Ao contrário do crossover que trata primeiramente o segmento de recursos, a mutação trata primeiramente o segmento de *proxy* de atividades (figura 8).

Um ou mais genes do segmento de *proxy* de atividade são escolhidos aleatoriamente para mutação.

É realizada uma verificação no *proxy* se todas as demandas deste gene já foram atendidas.

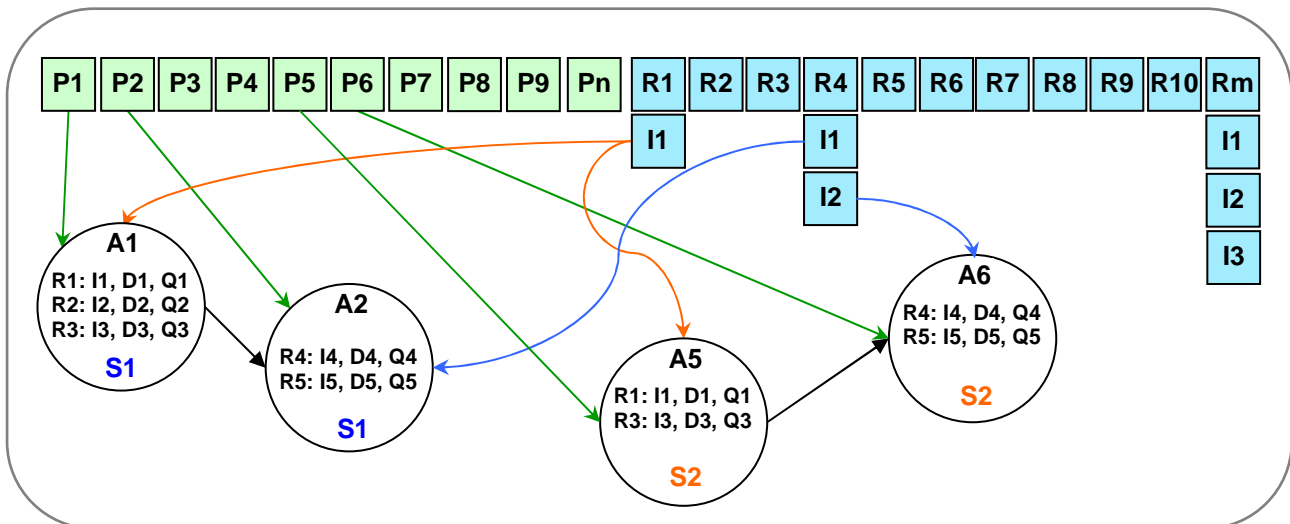


Figura 8 – Exemplos de Mutação ocorrendo (representação simplificada)

Caso existam demandas sem atendimento, é sorteada aleatoriamente uma demanda de recurso na atividade associada ao *proxy*, que como já foi apresentado anteriormente, contém informação de início e duração da demanda e quantidade de itens deste tipo de recurso. De posse desta demanda é buscado no segmento de recursos, o tipo de recurso correspondente a esta demanda.

O algoritmo tenta alocar o tipo de recurso para a demanda da atividade naquele período em que é necessário e na quantidade exigida de acordo com os seguintes passos:

1. Se não há nenhum item de recurso alocado para o tipo de recurso exigido, então cria o primeiro item de recurso, bem como sua lista para associar as atividades atendidas de acordo com o período.
2. Busca para cada item de recurso na sua lista de atividades atendidas um espaço (*slot*) para que a demanda da atividade seja atendida. Ou seja, o período de que a demanda necessita do recurso não pode coincidir com um período em que outra demanda já tenha alocado este item de recurso.
3. Caso haja espaço nesta lista de atividades atendidas para aquele item de recurso, então a demanda da atividade é registrada nesta lista para este item e termina a busca por *slots* nos itens de recurso (figura 8: R1 → I1 → A1, A5), ou
4. Se existir uma demanda por uma quantidade maior que uma unidade de um tipo de recurso, a busca pelos itens de recurso continua até conseguir *slots* para atender toda a quantidade de recursos da demanda.
5. Se não há nenhum *slot* na lista de atividades atendidas de um item de recurso para o período da demanda, então tenta o próximo item de recurso, até conseguir um *slot* vago (figura 8: R4 → I1 → A2; I2 → A6).
6. Novos itens de recurso serão criados para aquele tipo de recurso se os itens de recurso anteriores não suprirem a demanda.
7. Atualiza informação de quantidade de demandas atendidas no *proxy* correspondente à solicitação de alocação.

A mutação também pode funcionar ao contrário, ao invés de alocar recursos a uma demanda ela pode desalocar recursos previamente alocados. De forma a permitir que demandas de outras atividades possam escolher este novo espaço (*slot*).

Caso todas as demandas já tenham sido atendidas para a atividade daquele *proxy*, o operador de mutação pode escolher a próxima atividade no grafo de

atividades e realizar um processamento similar a este. Sendo o que ele irá realizar é sempre definido aleatoriamente de acordo com o contexto em que se encontra.

#### **4.4. Avaliação da População**

A população é avaliada a cada geração verificando se as demandas das atividades foram atendidas, bem como analisando a quantidade de todos os tipos de recursos que foram usados para atender estas demandas, e qual foi o custo total destas alocações.

Existe uma função para penalizar os indivíduos que não tiveram todas as suas demandas atendidas.

A função de avaliação classifica como melhores os indivíduos que atendem as demandas por recursos das atividades, e que possuam menor custo de contratação de equipamentos somado ao custo das diárias de sondas paradas.

##### **4.4.1. Penalização de Indivíduos**

A função de penalização (FP) é para penalizar os indivíduos que ainda não tiveram todas as suas demandas de atividade por recursos atendidas. Se não fosse utilizada esta função de penalização, um indivíduo que tivesse poucas demandas atendidas, também seria um indivíduo que consumiria poucos recursos, logo poderia haver uma tendência a diminuição do custo.

Como nesta fase do trabalho ainda não estamos utilizando o processo de atrasar as atividades para refletir o não atendimento de demandas de atividade no prazo, então não teríamos a priori como contabilizar o seu atraso. E mesmo se estivéssemos já usando este processo, o não atendimento a uma demanda durante a evolução não teria uma forma precisa de identificar o atraso real, pois não saberíamos quando esta atividade teria todas as suas demandas atendidas e só assim calcular este atraso. Logo também deveria ser utilizada uma forma artificial de penalizar estes indivíduos.

Desta forma, a função de penalização verifica o total de demandas por recursos de uma determinada atividade (verifica a partir do segmento de *proxy* de atividade do cromossomo) que está disponível na lista de demandas de uma atividade como visto anteriormente. E verifica o número de demandas atendidas

desta mesma lista. Se a subtração destes valores for 0 (zero), todas as demandas foram atendidas, então não adiciona nada para o computo do custo. Contudo, se alguma das demandas não foram atendidas, então contabiliza um valor positivo de penalização de acordo com a fórmula (FP):

$$\sum_{i=1}^n \sum_{j=1}^m A_i \rightarrow DN_j.quant\_rec \times [DN_j.valor\_diária\_rec] \cdot [1] \times mult$$

onde,  $A_i$  é uma atividade,  $DN_j$  é uma demanda não atendida,  $DN_j.quant\_rec$  é a quantidade de um determinado tipo de recurso da demanda,  $DN_j.valor\_diária\_rec$  é o valor da diária deste recurso e  $mult$  é o multiplicador a ser aplicado. No lugar do valor da diária podemos simplesmente utilizar a constante 1 para ficar independente do valor do recurso. De qualquer forma o valor do recurso é normalizado linearmente no intervalo  $0 < x \leq 1$ , para não haver muita discrepância entre um valor de recurso muito alto e um valor de recurso muito baixo.

#### 4.4.2. Avaliação do Custo

A função de avaliação, que verifica qual é o custo do conjunto de recursos alocados para as atividades é computada realizando o somatório de todos os recursos alocados (SRA) somado à função de penalização (FP), como segue (FA = SRA + FP):

SRA:

$$\sum_{i=1}^n \sum_{j=1}^m R_i \rightarrow I_j.valor\_diária\_rec$$

onde,  $R_i$  é um recurso e  $I_j$  é um item deste tipo de recurso.

O somatório de todos os recursos alocados (SRA) é determinado ao percorrer o segmento de Recursos do cromossomo, bem como seus itens de recurso para realizar o cálculo do custo dos recursos alocados para as atividades. Isto é feito desta forma pois assim temos como saber, recurso a recurso, para quais

atividades estão sendo alocados e em quais períodos de tempo, sem colisões e sem duplicações. Enquanto da forma de cálculo da função de penalização (FP) só tínhamos a visão parcial de cada atividade.

## 5 Resultados

Foram realizados alguns testes mais genéricos após o protótipo inicial ter ficado pronto. Os dados informados nos testes para o protótipo foram simplificados em relação aos dados reais, bem como alterados de forma a serem descaracterizados visto que são dados importantes.

### 5.1. Execução de Testes

Definimos como parâmetros para o algoritmo genético, os valores de número de gerações como sendo 100 iterações, população inicial de 50 indivíduos, taxa de crossover em torno de 80% (probabilidade de 0.8) e taxa de mutação em torno de 5% (probabilidade de 0.05). Está sendo utilizado Elitismo também, onde o melhor indivíduo de cada população é copiado para a população seguinte.

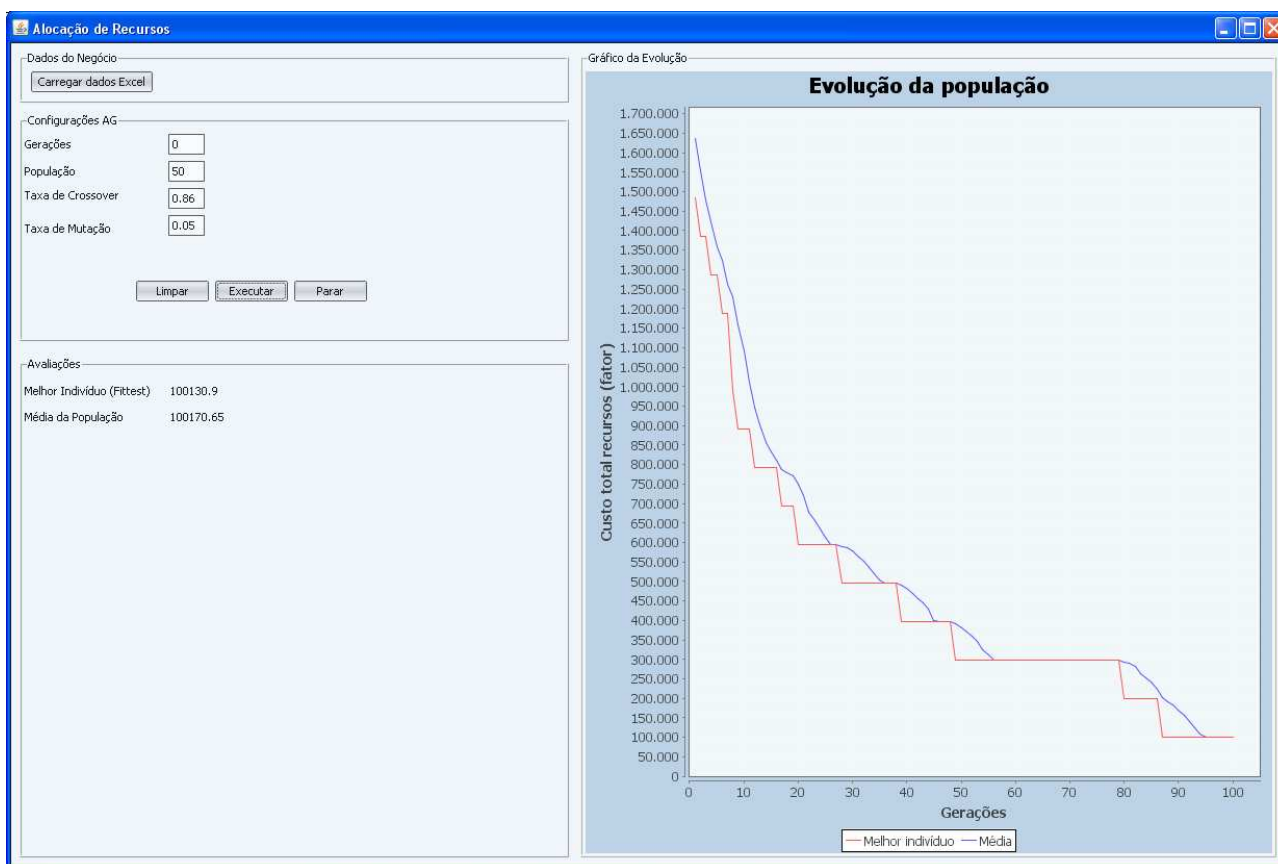


Figura 9 – Algoritmo genético após ter executado até terminar as 100 iterações

A figura 9 acima mostra um exemplo do algoritmo genético após ter sido executado. Cabe chamar a atenção para um ponto, o número de gerações é exibido como 0, pois ele foi decrescendo do número inicial de 100 indivíduos, logo ao término finalizou na última iteração e passou para 0. Cabe ressaltar também que o eixo y exibe o “Custo total recursos (fator)” que não é o custo real, mas um fator (é um *fitness*, uma aptidão) para definir o quanto está sendo avançado em direção a melhor solução. Como este é um problema de minimização, quanto menor for o custo melhor será a solução. Como foi explicado no capítulo 4, alguns indivíduos podem ser penalizados em determinadas condições.

Podemos verificar que com o *crossover* e a mutação o algoritmo tende a não ficar estacionado por muito tempo em um mesmo lugar, o período em que ficou mais tempo estagnado, foi entre as gerações de número 50 e 80, contudo conseguiu avançar em busca da melhor solução. E o elitismo permite que o algoritmo não retroceda uma vez que achou uma solução melhor que a anterior.

Como comparativo de eficácia do algoritmo genético foram realizados alguns testes utilizando uma forma de algoritmo de busca aleatória. Este algoritmo utiliza as mesmas restrições e validações do algoritmo genético, bem como o elitismo para levar o melhor indivíduo da população atual para a população seguinte.

Para criar este algoritmo de forma simplificada foi utilizado o mesmo código do algoritmo genético, porém com a taxa de *crossover* configurada para 0% (probabilidade de 0.0) e a taxa de mutação configurada para 100% (probabilidade de 1) de forma que fosse um algoritmo totalmente aleatório e que não utilizasse nenhum outro benefício dos algoritmos genéticos.

A figura 10 exibe o resultado de uma das execuções do algoritmo de busca aleatória utilizando os mesmos parâmetros que o algoritmo genético com exceção da taxa de *crossover* que é de 1 e taxa de mutação que é de 0.

Como pode ser visto nesta figura o algoritmo de busca aleatória tende a ficar bem mais tempo estagnado, como entre as iterações de número 50 e 100 e no geral ele não consegue chegar tão próximo do resultado como o algoritmo genético. Em comparações realizadas entre estes dois algoritmos o algoritmo genético é mais eficiente 30% em média em relação ao algoritmo de busca aleatória.



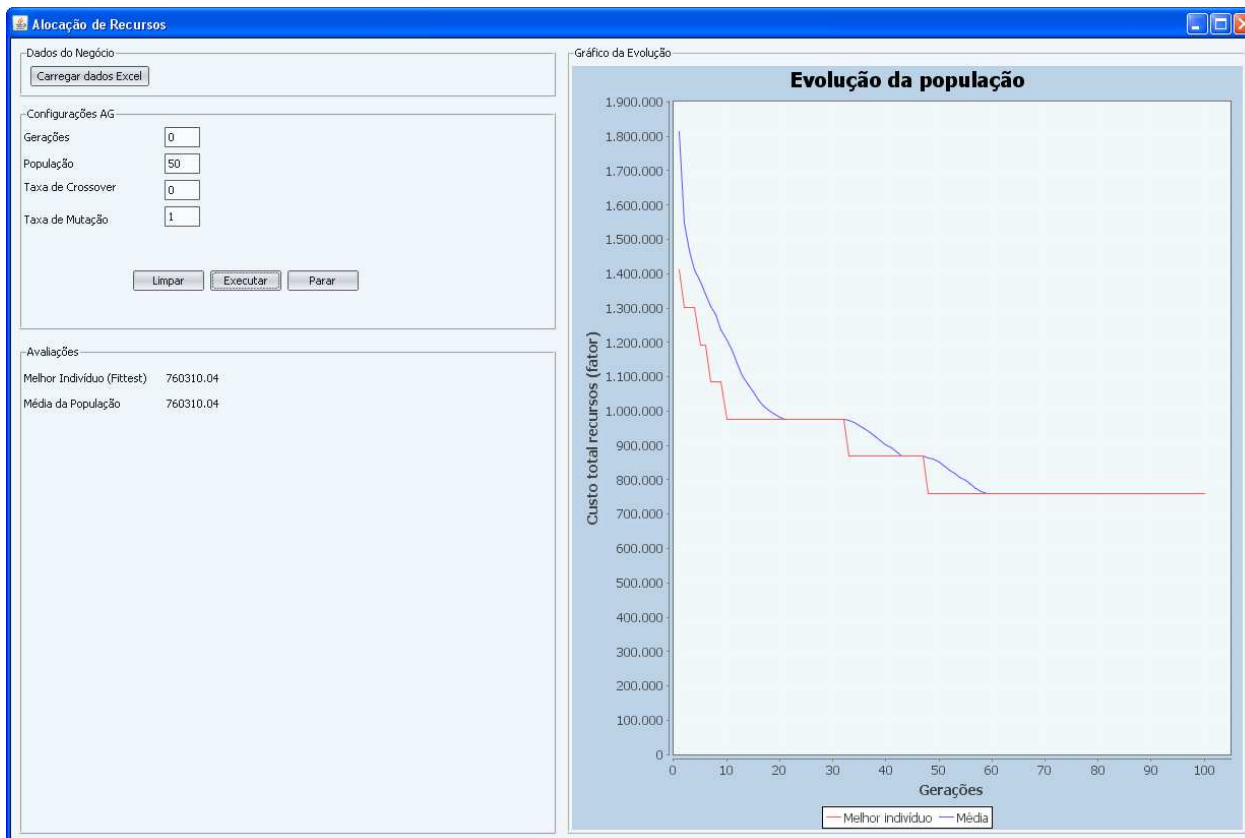


Figura 10 – Algoritmo de busca aleatória após ter executado até terminar as 100 iterações

## 5.2. Dados Utilizados

Como os dados sobre equipamentos são importantes, eles foram descaracterizados para serem apresentados aqui, logo não representam os dados reais da empresa.

A figura 11 apresenta um exemplo de listagem em forma textual de alguns itens do grafo de atividades. Podem ser destacados alguns itens como por exemplo o Recurso de código 221 que está em negrito nesta figura. Ele possui um valor de diária igual a '1.3' sendo que está sendo representado em milhares de reais. Alguns itens estão representados como null (ou seja ausência de valor), pois nesta representação não são informados para este caso.

```

Atividade [A| codigo=12122, nome=DMM 3, inicio=21/10/2010 00:00, duracao=6,
sonda=null, projeto=null, listaDemandaRecurso=[] |A]
Atividade [A| codigo=1444, nome=BARCO DE ESTIMULAÇÃO CA, inicio=24/08/2010
00:00, duracao=6, sonda=null, projeto=null,
listaDemandaRecurso=[DemandaRecurso [D| atividade=null, recurso=Recurso [R|
codigo=534, nome=BARCO DE ESTIMULAÇÃO, valorDiaria=30.6 |R],
quantidadeRecurso=1, inicioDemanda=24/08/2010 00:00, duracaoDemanda=6 |D]] |A]
Atividade [A| codigo=12245, nome=ABANDONO, inicio=01/10/2010 00:00, duracao=3,
sonda=null, projeto=null, listaDemandaRecurso=[] |A]
Atividade [A| codigo=222333, nome=COP - TH, inicio=01/09/2010 00:00,
duracao=30, sonda=null, projeto=null, listaDemandaRecurso=[DemandaRecurso [D|
atividade=null, recurso=Recurso [R| codigo=221, nome=CHAVE HIDRÁULICA,
valorDiaria=1.3 |R], quantidadeRecurso=1, inicioDemanda=01/09/2010 00:00,
duracaoDemanda=32 |D], DemandaRecurso [D| atividade=null, recurso=Recurso [R|
codigo=560, nome=ANALISADOR (CHAVE HIDRÁULICA), valorDiaria=0.03 |R],
quantidadeRecurso=1, inicioDemanda=01/09/2010 00:00, duracaoDemanda=32 |D],
DemandaRecurso [D| atividade=null, recurso=Recurso [R| codigo=1001,
nome=UNIDADE DE ARAME, valorDiaria=0.02 |R], quantidadeRecurso=1,
inicioDemanda=01/09/2010 00:00, duracaoDemanda=30 |D]] |A]
Atividade [A| codigo=22222, nome=ANM, inicio=01/10/2010 00:00, duracao=6,
sonda=null, projeto=null, listaDemandaRecurso=[DemandaRecurso [D|
atividade=null, recurso=Recurso [R| codigo=558, nome=CHAVE HIDRÁULICA,
valorDiaria=1.2 |R], quantidadeRecurso=1, inicioDemanda=01/10/2010 00:00,
duracaoDemanda=6 |D]] |A]
Atividade [A| codigo=333333, nome=CONTINGÊNCIA DA DOCAGEM (25%),
inicio=26/10/2010 00:00, duracao=24, sonda=null, projeto=null,
listaDemandaRecurso=[] |A]
Atividade [A| codigo=444444, nome=ABANDONO, inicio=19/04/2010 00:00,
duracao=7, sonda=null, projeto=null, listaDemandaRecurso=[] |A]
Atividade [A| codigo=445544, nome=DOCAGEM DE SONDA, inicio=28/06/2010 00:00,
duracao=120, sonda=null, projeto=null, listaDemandaRecurso=[] |A]

```

Figura 11 – Listagem de alguns dados do grafo de atividades e demandas

Ainda falta neste trabalho desenvolver uma forma de representação visual da saída uma vez que na forma textual é mais complicada a sua análise. Pode ser utilizado no futuro um gráfico de Gantt para representar as atividades e suas dependências. A ferramenta Jung utilizada para representar grafos também possui representações visuais que também podem ser utilizadas em trabalhos futuros.

## 6 Conclusões e Trabalhos Futuros

Os objetivos iniciais deste trabalho eram:

- O desenvolvimento de um protótipo de sistema para alocação de recursos utilizando Algoritmos Genéticos;
- A criação de um método escalável para alocação de recursos em atividades de sondas de perfuração;
- A definição de uma forma otimizada de estimar a quantidade de recursos para as atividades ao longo de um período de tempo;

Estes objetivos foram concluídos apesar de ter sido necessário definir uma versão mais simplificada do problema original em vista do tempo hábil para desenvolver e terminar este trabalho.

Alguns dos itens que foram excluídos foram: Os relacionamentos de dependências entre as atividades da forma Início-a-Início (II), Início-a-Término, Término-a-Início e Término-a-Término (TT) que não foram contemplados inicialmente neste trabalho, somente as dependências do tipo Término-a-Início entre as atividades foram contempladas nesta fase do protótipo.

Também foi eliminada a possibilidade de se atrasar uma atividade devido à indisponibilidade de um recurso. Desta forma mesmo que se gaste mais recursos as atividades não serão atrasadas.

Os testes iniciais mostraram que este algoritmo genético é uma solução viável em relação ao algoritmo de busca aleatória. Devendo ser realizados mais testes e análises mais robustas nestes algoritmos para provar sua eficiência.

Finalmente, é necessário que sejam exibidos os resultados de forma visual para facilitar a identificação de possíveis problemas, uma vez que a análise de resultados em forma textual é bem trabalhosa.

## 6.1. Trabalhos Futuros

Como atualmente este trabalho só realiza a saída de dados de forma textual, é necessário desenvolver uma forma de representação visual da saída uma vez que na forma textual é mais complicada a sua análise. Poderá ser criado em implementações futuras um gráfico de Gantt para representar as atividades e suas dependências. A ferramenta Jung utilizada para representar grafos também possui representações visuais que podem ser utilizadas em trabalhos futuros para complementar as informações deste gráfico de Gantt.

Estender a arquitetura e o protótipo para atenderem os relacionamentos de dependências entre as atividades da forma Início-a-Início (II), Início-a-Término, Término-a-Início e Término-a-Término (TT) que não foram contemplados nesta fase atual do protótipo.

Analisar ou implementar a capacidade das atividades poderem atrasar em relação a possibilidade de se alugar menos recursos naquele ano, verificando e optando pela minimização do custo total.

Analisar e talvez acrescentar funcionalidades para avaliar de forma diferenciada (pesos diferentes) o atendimento a campos mais produtivos em detrimento de menos produtivos.

Talvez uma necessidade de trabalho futuro possa ser a alocação de pessoas capacitadas a atenderem determinadas atividades, pois cada um pode trabalhar em uma área específica e a demanda por essas pessoas pode aumentar à medida que aumenta a quantidade de projetos ou sondas. O tempo gasto para treinamento de pessoal pode não ser pequeno, vide os profissionais da área de engenharia que geralmente permanecem em treinamento por períodos de mais ou menos um ano após sua contratação e ainda podem realizar outros cursos específicos que duram meses.

Finalmente, o foco deste trabalho foi a alocação de recursos em atividades de projetos de intervenção em poços, contudo podemos pensar em realizar conjuntamente com este trabalho algo bem maior e que está relacionado que é o seqüenciamento de projetos e a definição do cronograma de sondas. Este conjunto de problemas talvez possa ser solucionado utilizando um algoritmo genético co-evolutivo que atribua diferentes prioridades ou pesos a cada tipo de problema.

## Referências Bibliográficas

Blazewicz, J.; Dror M.; Weglarz J. **Mathematical programming formulations for machine scheduling: A survey**. European Journal of Operations Research 51, 283 - 300, 1991.

Cruz, A. V. A.; Pacheco, M. A. C.; Vellasco, M. M. R. **Otimização de Planejamentos com Restrição de Precedência usando Algoritmos Genéticos e Co-Evolução Cooperativa**. Rio de Janeiro, 2003. 79p. Dissertação de Mestrado – Departamento de Engenharia Elétrica, Pontifícia Universidade Católica do Rio de Janeiro.

ECJ - A Java-based Evolutionary Computation Research System. Disponível em: <<http://cs.gmu.edu/~eclab/projects/ecj>>. Acesso em 06 de dezembro 2011.

EvoJ. Disponível em: <<http://evoj-frmw.appspot.com/index.html>>. Acesso em 06 de dezembro 2011.

Gamma, E.; Helm, R.; Johnson, R.; Vlissides, J. **Design Patterns: Elements of Reusable Object-Oriented Software**. Addison-Wesley, 1995.

Haupt, R. L.; Haupt, S. E. **Practical genetic algorithms**. John Wiley & Sons, 2ª ed., 2004.

Holland, J. H. **Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence**. A Bradford Book, 1992. (MIT Press First Edition, 1975)

Ipate, F.; Lefticaru, R. **Genetic Model based Testing: a Framework and a Case Study**. Romanian Journal of Information Science and Technology, 11(3), pp. 209–227, 2008.

JAGA. Disponível em: <<http://www.jaga.org>>. Acesso em 06 de dezembro 2011.

Jenes. Disponível em: <<http://jenes.cislab.org>>. Acesso em 06 de dezembro 2011.

JGAP - Java Genetic Algorithms Package. Disponível em: <<http://jgap.sourceforge.net>>. Acesso em 06 de dezembro 2011.

JUNG - Java Universal Network/Graph Framework. Disponível em: <<http://jung.sourceforge.net>>. Acesso em 06 de dezembro 2011.

Lazzarine, A.L.; Oliveira, J.C.; Marcellino, F.J.M.; Accioly, R.M.S. **Sistema de Nivelamento de Recursos Críticos**. IV Seminário SPCAD, 2008.

Lefticaru, R.; Ipate, F. **Automatic State-Based Test Generation Using Genetic Algorithms**. Proceedings of the Ninth International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC 2007), IEEE Computer Society, pp. 188-195, 2007.

Lefticaru, R.; Ipate, F.; Tudose, C. **Automated Model Design using Genetic Algorithms and Model Checking**. Proceedings of the 4th Balkan Conference in Informatics (BCI'09), IEEE Computer Society, 2009

Mazzini, F.F. **Problema de Scheduling de Projetos com Restrição de Recursos**. Relatório Técnico, TIC/TIC-SERV/SPO, Petrobras, 2007.

Mazzini, F.F.; Accioly, R.M.S.; Vasconcellos, R.V.J.C. **Dimensionamento de Equipamentos Críticos da Cadeia de Suprimento da Perfuração e Completação de Poços**. Anais do XLII Simpósio Brasileiro de Pesquisa Operacional, Bento Gonçalves, Brasil, 2010.

Metsker, S. J. **Design Patterns Java Workbook**. Addison-Wesley, 2002.

Michalewicz, Z. **Genetic Algorithms + data structures = evolution programs**. Springer-Verlag, 1996.

Netto, P. O. B. **Grafos: Teoria, Modelos, Algoritmos**. Edgard Blücher, 4<sup>a</sup> Ed, 2006

Romera, A. J.; Beukes, P.; Clark, C.; Clark, D.; Levy, H.; Tait, A. **Use of a pasture growth model to estimate herbage mass at a paddock scale and assist management on dairy farms**. Computers and electronics in agriculture, Volume 74, issue 1, pp. 66-72, October 2010.

Rothlauf, F. **Representations for Genetic and Evolutionary Algorithms**. Springer-Verlag, 2006.

Shyshou, A. **A simulation study of the fleet sizing problem arising in offshore anchor handling operations**. European Journal of Operations Research, 2010.

Stelting, S.; Maassen, O. **Applied Java Patterns**. Prentice Hall, 2001.

Vasconcellos, R.V.J.C.; Ferreira, V.J.M. **Algoritmo genético para o problema de scheduling de projetos com restrição de recursos: Uma aplicação em operações em poços de petróleo**. Anais do XXXVIII Simpósio Brasileiro de Pesquisa Operacional, 2006.

Wall, M. B. **A Genetic Algorithm for Resource-Constrained Scheduling**. Cambridge, 1996. 62p. Tese de Doutorado – Department of Mechanical Engineering, Massachusetts Institute of Technology.

Watchmaker Framework. Disponível em:  
<<http://watchmaker.uncommons.org>>. Acesso em 06 de dezembro 2011.