

# Análise de Comunalidade em Sistemas Espaciais Através de Algoritmo Genético

Anderson Cattelan Zigiotta<sup>1</sup>, Roberto d'Amore<sup>2</sup>

<sup>1</sup>Instituto de Aeronáutica e Espaço – Praça Eduardo Gomes, 50 – CEP 12228-904 – São José dos Campos – Brasil

<sup>2</sup>Instituto Tecnológico de Aeronáutica – Praça Eduardo Gomes, 50 – CEP 12228-900 – São José dos Campos – Brasil

**Resumo** — Comunalidade permite que dois ou mais itens de um sistema sejam substituídos por um item comum, diminuindo custos de pesquisa e desenvolvimento, manufatura e operação. Isto é especialmente interessante em sistemas espaciais, os quais possuem baixo volume de produção e longo ciclo de projeto. Itens comuns, no entanto, geralmente têm desempenho inferior e excesso de funcionalidade em relação aos itens especializados. Para auxiliar o projetista na decisão sobre quais itens devem ser agrupados, alguns métodos podem ser utilizados. Este trabalho apresenta um algoritmo genético para resolver o problema da análise de comunalidade em sistemas espaciais. São descritos a codificação das soluções e os operadores de cruzamento e mutação adequados ao problema. Aplicado a um exemplo que analisa as especificações dos motores elétricos da Estação Espacial Internacional, e comparado a um algoritmo de agrupamento hierárquico, o algoritmo proposto encontrou os melhores resultados, permitindo ainda utilizar restrições e múltiplos objetivos independentes quando necessário.

**Palavras-Chave** — Sistemas espaciais, Comunalidade, Algoritmo Genético.

## I. INTRODUÇÃO E MOTIVAÇÕES

Os sistemas e equipamentos para uso espacial possuem características que os diferenciam dos demais. Uma destas características é o baixo volume de produção. De maneira geral, a quantidade de produtos e missões nesta área é bastante reduzida se comparada com a área comercial. Um equipamento pode ser produzido para uma única missão, como no caso de sondas e satélites, ou até para realizar algumas missões por ano, como no caso de veículos lançadores, de sondagem e cargas úteis. O baixo volume de produção impede que o equipamento seja beneficiado por economia de escala, ou pelo aprimoramento do processo de produção que ocorre naturalmente ao longo do tempo.

Outra característica é que os equipamentos espaciais possuem um ciclo de projeto extenso e demorado devido ao tipo de missão, que requer alta confiabilidade. Este ciclo começa em uma especificação criteriosa, seguida de diversas etapas de projeto e revisões críticas, além de uma série de ensaios funcionais e ambientais. Como resultado, além de aumento do custo, cada equipamento demanda um tempo razoável para ser considerado qualificado para a missão.

O custo de um sistema é composto de quatro componentes: custo de pesquisa e desenvolvimento – P&D; custo de produção; custo operacional e custo de descarte. Enquanto os custos de produção variam proporcionalmente com o tamanho do lote produzido, os custos de P&D geralmente são não recorrentes. Assim, devido à pequena quantidade produzida e ao longo ciclo de testes, os custos de P&D são excepcionalmente grandes para os sistemas

aeroespaciais. Uma possível solução para minimizar este problema é a utilização de comunalidade, que é a propriedade de dois ou mais itens possuírem características comuns. Itens com alta comunalidade podem ser trocados entre si ou substituídos por um único item comum. A comunalidade pode existir nos diversos níveis de um projeto.

Dentre os possíveis modelos de auxílio ao projetista na decisão sobre quais itens podem ser tornados comuns, um dos mais promissores consiste em tratar a análise de comunalidade como um problema de particionamento [1]. No modelo, as possíveis soluções são representadas por partições, onde os elementos pertencentes a uma mesma partição são agrupados, ou seja, tornados comuns. Desta maneira pode-se utilizar um algoritmo que encontre a melhor partição, isto é, aquela que possua o menor custo. Esta abordagem foi empregada em um problema na área espacial para encontrar a solução de comunalidade para equipamentos da Estação Espacial Internacional – EEI – usando, para isso, um algoritmo de agrupamento hierárquico.

O algoritmo mostra-se bastante eficiente e rápido. No entanto, ele possui algumas desvantagens. Uma vez realizados, os agrupamentos não podem ser refeitos o que impede o algoritmo de explorar outras regiões do espaço de soluções. O algoritmo na forma apresentada não utiliza restrições tampouco possibilita a otimização com vários objetivos independentes e separados. Estas características são desejáveis quando se deseja aplicar tal método no estudo de comunalidade de sistemas mais complexos, como a arquiteturas de instrumentação e controle de foguetes de sondagem e suas cargas úteis por exemplo.

Com o objetivo de possibilitar o uso de restrições e múltiplas funções objetivo, neste trabalho é proposto o uso de algoritmos genéticos – AGs – para resolver o problema de particionamento aplicado à escolha da comunalidade em sistemas espaciais. No entanto, antes de aplicar o algoritmo a problemas com estas características, inicialmente é necessário verificar a viabilidade de se utilizar AGs para resolver este tipo de problema. Para tanto, é preciso haver dados para fins de comparação. Com essa finalidade o algoritmo proposto é testado com o problema dos motores da EEI existente na literatura.

## II. COMUNALIDADE EM SISTEMAS

A comunalidade pode reduzir os custos de P&D, ao substituir diferentes equipamentos especializados por um único equipamento comum. Também diminui os custos de produção, através da aquisição de lotes maiores de componentes e do ganho de produtividade devido à curva de aprendizado. Outra vantagem é que a diminuição da

variedade de equipamentos e, por conseguinte, de procedimentos, reduz os custos de operação, incluindo aqueles associados a estoque, documentação e treinamento. Para componentes é possível substituir resistores ou capacitores por outros com valores próximos, de forma a reduzir a quantidade de itens únicos. Isso diminui os custos de aquisição dos componentes, pois o lote é maior, e também o custo de estoque. Em nível superior, no sistema pode existir um equipamento comum que incorpore as funções exercidas por dois ou mais equipamentos especializados. A comunalidade de sistemas e subsistemas pode ser exemplificada pela existência de um mesmo sistema de suprimento de energia para uma família de satélites.

O uso de comunalidade, no entanto, possui desvantagens. Quando dois itens A e B são reunidos no item comum AB, este deve satisfazer os requisitos funcionais dos dois itens especializados. Logo, ele terá algumas funções ou componentes em excesso em relação ao item A e em relação ao item B. Esta característica é chamada excesso de funcionalidade e, na prática, pode ser considerada como aumento de massa, volume, consumo de energia e taxa de falhas. Além disso, de maneira geral, um item comum possui menor desempenho que os itens especializados na realização das suas respectivas funções.

Assim, cabe ao projetista do sistema considerar estas vantagens e desvantagens e escolher as peças ou equipamentos para os quais é proveitoso utilizar comunalidade. Para sistemas com poucos equipamentos, ou equipamentos com poucos componentes, essa decisão pode ser feita manualmente. Por exemplo, em uma análise com apenas dois itens, A e B, existem duas alternativas, ou partições, possíveis: (a) desenvolver e produzir os dois itens, A e B, resultando na solução sem comunalidade  $\{\{A\},\{B\}\}$ ; (b) desenvolver e produzir um único item AB, o qual realiza as funções dos dois itens substituídos, resultando na solução de comunalidade total  $\{\{AB\}\}$ . O número de soluções possíveis para o problema da comunalidade é igual ao número de maneiras em que um conjunto de  $n$  elementos pode ser particionado em conjuntos não vazios, e é chamado número de Bell –  $B(n)$ .

O número de Bell para 25 objetos é aproximadamente  $4,6 \cdot 10^{18}$  e para 40 objetos  $1,6 \cdot 10^{35}$ . Assim, o problema é combinatório e encontrar a melhor solução através da enumeração, ou busca exaustiva, não é viável a partir de certo número de itens. Portanto, torna-se necessária uma ferramenta de auxílio ao projetista na decisão sobre o nível de comunalidade a ser empregado.

### III. MODELO PARA ANÁLISE DE COMUNALIDADE DA EEI

A abordagem empregada em [2] para encontrar a solução de comunalidade para equipamentos da Estação Espacial Internacional usou um algoritmo de agrupamento hierárquico baseado no trabalho de Ward [3]. Este algoritmo consiste em uma técnica aglomerativa que agrupa dois elementos em cada passo, baseado em uma matriz de dissimilaridades. Esta contém as mudanças na função objetivo decorrentes do agrupamento dos itens dois a dois. O objetivo é diminuir a

variedade de itens, substituindo componentes diferentes por um comum que atenda os mesmos requisitos, daí o termo “análise de comunalidade”.

No exemplo a ser tomado como referência, estudos preliminares indicavam a necessidade de 27 tipos diferentes de motores elétricos para o sistema EEI. A reunião de motores semelhantes em um agrupamento resulta nas especificações de um motor que pode substituir todos aqueles pertencentes a este agrupamento. A função objetivo contabiliza os ganhos de custos de P&D e de produção, bem como as perdas por excesso de funcionalidade, transformadas em unidade monetária. Desta maneira, estes dois objetivos conflitantes são reunidos em uma única função através de uma combinação linear, cujos pesos são tomados a partir de um banco de dados da NASA. O custo da solução sem comunalidade é igual a 4058,93 e, aplicando o algoritmo de agrupamento hierárquico ao exemplo, a melhor partição encontrada corresponde a 14 motores distintos ao custo de 3418,59, uma redução de aproximadamente 15 %.

O algoritmo de agrupamento hierárquico, apesar de sua alta velocidade, apresenta algumas deficiências, principalmente quando se cogita utilizar o método para a análise de sistemas mais complexos, como arquiteturas de equipamentos eletrônicos embarcados em veículos de sondagem e suas cargas úteis.

Primeiramente, eles trabalham com uma única função objetivo, e o problema possui no mínimo dois objetivos conflitantes: reduzir custos de P&D e de produção; e reduzir o excesso de funcionalidade. No exemplo apresentado isto é resolvido através de uma função objetivo formada por uma combinação linear entre os objetivos conflitantes. Esta solução exige que os pesos relativos sejam conhecidos e confiáveis, o que é particularmente difícil em sistemas em que há pouca experiência de projetistas de sistemas e com baixa cadência de produção. Outra dificuldade com algoritmos hierárquicos, ao menos na sua forma canônica, é a ausência de restrições. Restrições são importantes para garantir a viabilidade das soluções. Por exemplo, uma possível restrição seria um valor máximo absoluto para o volume de um equipamento, ou então um valor máximo relativo para o consumo de energia. Ao colocar restrições no agrupamento hierárquico corre-se o risco de parar o seu funcionamento antes que uma solução boa seja atingida. Por fim, ao utilizarem unicamente decisões locais e irreversíveis, estes algoritmos podem não conseguir atingir uma área grande do espaço de soluções e, por consequência, perder a oportunidade de encontrar soluções boas ou mesmo ótimas.

### IV. ALGORITMO GENÉTICO PARA O PROBLEMA DA ANÁLISE DE COMUNALIDADE

Algoritmos evolutivos também são utilizados para resolver problemas de agrupamento [4]. E dentre esta classe destacam-se os algoritmos genéticos [5], os quais utilizam populações de soluções que se combinam e são selecionadas, evoluindo para melhores resultados. Dentre as características desta classe de algoritmos, são de especial interesse, uso de restrições e possibilidade de se trabalhar com objetivos independentes. Os AGs são métodos de busca e otimização

inspirados no processo de evolução dos organismos vivos. Cada indivíduo de uma população corresponde a uma solução do problema, sendo codificada em uma estrutura de dados chamada *cromossomo*.

O algoritmo começa com uma população inicial formada por certo número de indivíduos, ou cromossomos. Esta população é avaliada e a cada indivíduo é atribuído um valor de *aptidão*, o qual representa a qualidade desta solução. De maneira geral, o valor desta aptidão é o valor da própria função objetivo do problema, podendo ser ranqueada ou escalonada. Desta população escolhem-se os melhores indivíduos que poderão gerar descendentes. Existem várias maneiras de selecionar os indivíduos-pais. De maneira geral utiliza-se um processo aleatório, onde os indivíduos mais aptos têm maior chance de serem selecionados. Permite-se assim que alguns indivíduos menos aptos consigam passar pela seleção, a fim de garantir a diversidade da próxima população e evitar a convergência prematura para um valor ótimo localmente. Os indivíduos-pais geram membros da próxima população através de um processo de cruzamento, onde trocam características genéticas. Este processo ocorre com uma probabilidade alta. Os descendentes, assim, possuem características combinadas dos dois indivíduos-pais. A fim de garantir a diversidade da população e permitir outras regiões do espaço de soluções sejam visitadas, é aplicado aos descendentes um operador de *mutação*. Este operador altera aleatoriamente um parâmetro, ou *gene*, do indivíduo, e é aplicado com uma probabilidade baixa.

A nova geração passa novamente pelos processos de avaliação, seleção, cruzamento e mutação. O ciclo é repetido até que uma determinada condição de parada seja atingida. Esta condição pode ser o número máximo de gerações ou o número de ciclos sem melhoria da solução.

Na Tabela I é descrito o pseudocódigo que resume o funcionamento de um algoritmo genético canônico.

**TABELA I PSEUDOCÓDIGO DE UM ALGORITMO GENÉTICO CANÔNICO.**

Seja Pop(k) a população de cromossomos na geração k. $k \leftarrow 0$ inicializa Pop(k) avalia Pop(k) <b>enquanto</b> o critério de parada não for satisfeito <b>faça</b> $k \leftarrow k + 1$ seleciona Pop(k) a partir de Pop(k-1) aplica cruzamento sobre Pop(k) com probabilidade $P_{cross}$ aplica mutação sobre Pop(k) com probabilidade $P_{mut}$ avalia Pop(k) <b>fim enquanto</b>
---

Para empregar um AG na solução de um problema qualquer, deve-se definir: a codificação: como as soluções do problema podem ser representadas em um cromossomo; a avaliação da aptidão: se igual à função objetivo, se através de escalonamento ou ranqueamento; a seleção: se somente os melhores indivíduos, através de roda de roleta, amostragem estocástica universal, torneio, ou outro método; o operador cruzamento e sua probabilidade; o operador mutação e sua probabilidade; outros parâmetros do algoritmo, como o critério de parada e taxa de substituição de indivíduos na população. A definição de muitos destes parâmetros ocorre

por experimentação e os valores obtidos para um problema muitas vezes não se aplicam em outro.

### V. ALGORITMO GENÉTICO PARA ANÁLISE DE COMUNALIDADE DA EEI

A fim de verificar a viabilidade no uso de AGs para o problema de comunalidade, utilizou-se o mesmo conjunto de dados dos motores elétricos da EEI, bem como a mesma função objetivo.

Para codificação das soluções, ou cromossomos, foi utilizado o conceito de RGF – *Restricted Growth Function*, como no algoritmo genético proposto por Tucker et. al [6]. Uma RGF é uma função  $f : [n] \rightarrow [n]$  tal que  $f(1) = 1$  e  $f(i + 1) \leq \max\{f(1), \dots, f(i)\} + 1$ . A RGF pode representar uma partição em  $m \leq n$  grupos, em que ‘1’ por convenção pertence ao primeiro grupo,  $i$  pertence ao  $f(i)$ -ésimo grupo e  $\max\{f(1), \dots, f(n)\} = m$ .

Desta maneira, [1,1,2,3,2,4] é uma RGF que representa a partição  $\{\{A,B\},\{C,E\},\{D\},\{F\}\}$  e [1,2,2,3,4,3] representa a partição  $\{\{A\},\{B,C\},\{D,F\},\{E\}\}$ , existindo uma relação unívoca entre uma partição e sua representação. Já [1,4,3,2,3,1] e [2,1,2,3], por exemplo, não são RGFs. Outro algoritmo genético que usa a mesma representação é o proposto por Du et. al. [7], embora a RGF seja chamada de RGS – *Restricted Growth String*.

A inicialização da população foi feita através da geração de cromossomos aleatórios. A seleção dos cromossomos para a população seguinte foi realizada através de torneio com dois competidores. Neste método um número de cromossomos é escolhido aleatoriamente da população e o melhor deles é transferido para a população intermediária. A própria função objetivo foi utilizada para avaliação. Ao algoritmo canônico foi acrescentado elitismo, o qual consiste em garantir que os melhores cromossomos sejam transferidos para a população seguinte.

Para o operador de cruzamento foram testadas quatro possibilidades.

- i. Cruzamento em um ponto: é escolhido um ponto do cromossomo a partir do qual os pais trocam alelos, é necessário retificar o cromossomo resultante para atender às regras de formação da RGF.
- ii. Cruzamento uniforme: uma máscara de elementos binários é gerada, dependendo do valor de cada posição, os filhos herdam os alelos de um ou de outro cromossomo-pai.
- iii. Cruzamento RGF: descrito em [6] consiste em gerar um “caminho” de funções RGF entre os dois cromossomos-pai e escolher aleatoriamente indivíduos pertencentes a este caminho para a próxima população.
- iv. Cruzamento LLE: adaptado de [8] realiza um “cruzamento de grupos” derivado da codificação em lista encadeada.

Já o operador mutação é praticamente um acordo na literatura e consiste em, aleatoriamente, realizar uma das seguintes funções: (a) um grupo é unido a outro; (b) um grupo é dividido em dois; (c) um elemento de um grupo é movido para outro.

Os parâmetros do algoritmo foram obtidos através de experimentação. Um tamanho da população próximo ao tamanho do cromossomo, no exemplo 28 indivíduos, é suficiente para obter diversidade sem aumentar excessivamente o tempo de execução. Para facilitar a implementação do algoritmo o tamanho da população deve ser par. O critério de parada utilizado foi o do número máximo de gerações. Como regra geral, adotou-se o número de gerações igual a dez vezes o tamanho da população, pois observou-se que após este número de gerações raramente ocorria uma melhora no resultado. De maneira semelhante, ou seja, empiricamente, foram definidas as taxas de cruzamento em 0,30 e de mutação em 0,70 como aquelas que resultavam no melhor compromisso entre velocidade de execução e qualidade do resultado.

Partindo dessas definições é possível testar a aplicabilidade de algoritmos genéticos para resolver o problema. A Fig. 1 mostra os valores para as funções objetivo do melhor cromossomo e da média dos cromossomos da população em função da geração. Para este exemplo foi utilizado o cruzamento LLE com taxa 0,70 e taxa de mutação igual a 0,30, 28 indivíduos na população, 2 cromossomos elite, 280 gerações.

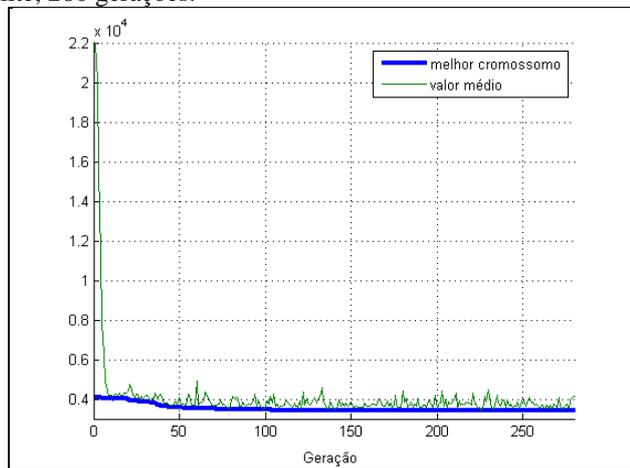


Fig. 1. Evolução da melhor solução e da média da população para o algoritmo genético.

O resultado do melhor cromossomo é igual ao obtido pelo método do agrupamento hierárquico. Devido ao tamanho relativamente pequeno do problema e após executar o algoritmo várias vezes, pode-se admitir que provavelmente o valor seja ótimo, isto é, corresponde ao mínimo da função objetivo. Isto mostra a viabilidade em se utilizar algoritmos genéticos para este problema.

O valor da função objetivo ao longo das gerações demonstra que o algoritmo converge rapidamente e seu melhor cromossomo praticamente estaciona por um longo tempo. Eventualmente uma solução melhor é encontrada.

O algoritmo genético é probabilístico. Assim, cada vez que ele é executado pode-se obter um resultado diferente. Com a finalidade de verificar esta variação, bem como comparar os diferentes operadores de cruzamento, executou-se o algoritmo genético cinquenta vezes com cada operador. As Fig. 2, 3, 4 e 5 mostram a distribuição de frequência das respostas obtidas para os operadores

um ponto, uniforme, RGF e LLE, respectivamente. Os outros parâmetros do AG foram mantidos. Nesta comparação, os melhores resultados foram obtidos pelos operadores cruzamento em um ponto e LLE.

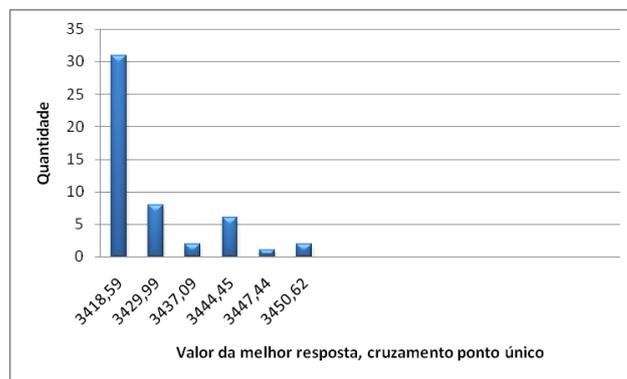


Fig. 2. Resultado do AG com operador cruzamento em um ponto.

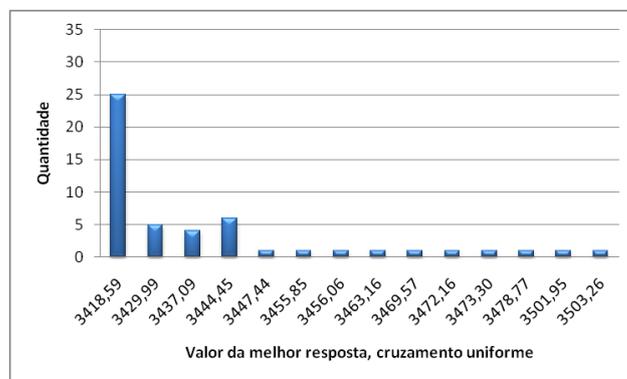


Fig. 3. Resultado do AG com operador cruzamento uniforme.

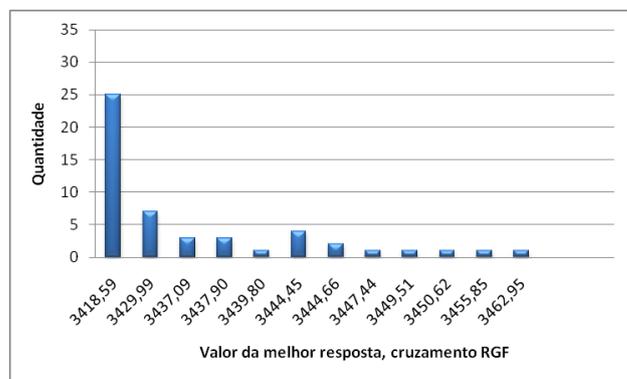


Fig. 4. Resultado do AG com operador cruzamento RGF.

Com a finalidade de estender esta análise para problemas com outros tamanhos, foram geradas sinteticamente três instâncias de problemas: “Mot75” com 75 motores; “Mot100a” e Mot100b” com 100 motores. Os tamanhos das populações foram de 76, 100 e 100 indivíduos e o número de gerações foi igual a 760, 1000 e 1000, respectivamente. Os demais parâmetros foram mantidos. Os resultados são listados na Tabela II.

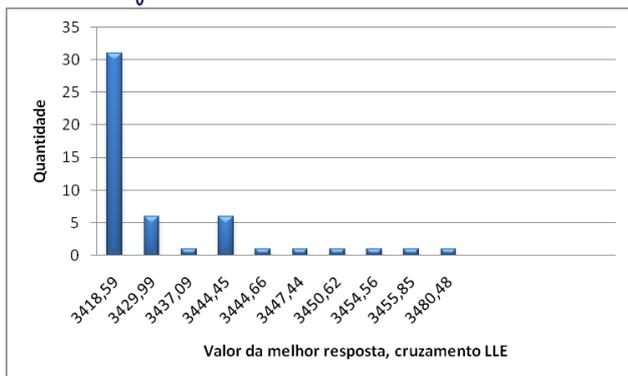


Fig. 5. Resultado do AG com operador cruzamento LLE.

Inicialmente são mostrados os custos para os sistemas sem comunalidade. Na segunda linha de dados aparecem os custos das soluções obtidas através de agrupamento hierárquico. Em seguida, são apresentados os custos das soluções resultantes do algoritmo genético proposto, utilizando-se dois tipos de operador cruzamento: em um ponto e LLE. Os resultados correspondem aos melhores e piores cromossomos e à média de todos os indivíduos existentes na última geração. Os melhores resultados estão destacados.

TABELA II RESULTADO PARA INSTÂNCIAS COM 75 E 100 MOTORES.

Instância	Mot75	Mot100a	Mot100b
<b>Custo sem comunalidade</b>	12822,14	73544,06	40666,94
<b>Agrupamento hierárquico</b>	9973,81	<b>66235,78</b>	<b>33819,90</b>
<b>AG com cruzamento um ponto</b>	<b>melhor</b>	<b>9967,33</b>	<b>66235,78</b>
	<b>média</b>	9997,31	66372,23
	<b>pior</b>	10057,89	66526,12
<b>AG com cruzamento LLE</b>	<b>melhor</b>	<b>9967,33</b>	<b>66235,78</b>
	<b>média</b>	9985,04	66336,67
	<b>pior</b>	10000,85	66560,11

## VI. CONCLUSÕES E CONTINUAÇÃO DO TRABALHO

Neste trabalho foi proposto um algoritmo genético de agrupamento para solucionar o problema de comunalidade em sistemas espaciais. Ele demonstrou efetividade ao encontrar boas soluções para o problema (não se pode garantir que sejam ótimas). A representação utilizada para o cromossomo foi do tipo RGF e o melhor operador cruzamento encontrado a partir de testes com um problema exemplo foi o do tipo LLE. Com esta configuração encontrou-se os melhores resultados para os problemas testados. Desta maneira, pode-se concluir que algoritmos genéticos com estas configurações podem ser utilizados para resolver o problema de comunalidade. Na continuação do trabalho, pretende-se utilizar este tipo de algoritmo para a definição de um sistema de eletrônica embarcada multimissão e multifunção em veículos de sondagem e cargas úteis. Para este problema serão utilizados objetivos separados e restrições de projeto, tais como volume e consumo de energia máximos.

Em relação ao algoritmo de agrupamento hierárquico, o qual serviu de referência, o AG proposto possui algumas vantagens e desvantagens. Os algoritmos genéticos possuem operadores que permitem explorar amplamente o espaço de soluções e conseguem avaliar requisitos de restrições, os quais poderiam inviabilizar um algoritmo hierárquico. Ainda, o fato da função objetivo ser transparente para o algoritmo permite que qualquer modelo possa ser utilizado, seja linear ou não. Outra vantagem é a possibilidade de se buscar a melhor solução considerando múltiplos objetivos separados, não necessitando de combinação entre objetivos, cujos pesos relativos são de difícil definição. Estas características são importantes para utilizar o método na análise de sistemas mais complexos.

Além disso, algoritmos baseados em população fornecem não apenas uma, mas várias soluções. Deste modo, o projetista pode analisar um grupo maior de soluções da população final, aplicando sua experiência para avaliar aspectos de difícil modelamento matemático

Por outro lado, o uso de algoritmos genéticos para agrupamento apresenta alguns problemas para encontrar boas soluções à medida que a dimensão do problema cresce, tornando necessário executar o algoritmo diversas vezes para aumentar a chance de encontrar uma boa solução. Os AGs ainda têm a desvantagem de executarem em um tempo maior que o algoritmo de agrupamento hierárquico. Esta diferença, de cerca de uma ordem de grandeza nos exemplos testados, pode ser explicada em parte pela linguagem em que cada algoritmo foi implementado. Enquanto o algoritmo de agrupamento foi executado a partir do código original em Turbo Pascal, o algoritmo genético foi interpretado em MATLAB. Outros fatores, como o estilo de escrita do código, também podem influenciar o tempo de execução. De qualquer maneira, para problemas relacionados a planejamento admite-se um tempo maior de execução do que para problemas que necessitam de resposta rápida, como o caso de sistemas em tempo real.

Outro possível problema deste tipo de algoritmo é a necessidade de ajustar vários parâmetros, como taxas de cruzamento e mutação, critério de parada, tipo de seleção, os quais influenciam não apenas o tempo de execução do algoritmo, mas também seu desempenho. Inicialmente estes parâmetros foram ajustados empiricamente, através de experimentações. Futuramente pretende-se utilizar técnicas como Delineamento de Experimentos para avaliar todos os parâmetros de forma simultânea.

O valor da função objetivo ao longo das gerações mostra que há uma rápida convergência do espaço de soluções e melhora acentuada da função objetivo e, em seguida, poucas melhoras ocorrem. Esta parece ser uma característica dos algoritmos de agrupamento em geral, os quais inicialmente conseguem agrupar facilmente os itens e então somente realizam melhoras marginais através de mudanças de itens entre os agrupamentos. A partir desta característica, pode-se propor um algoritmo com taxas de mutação e cruzamento variáveis: baixas ou nulas no início da execução e mais altas quando o valor da melhor função objetivo encontrada na população estabilizar. As baixas taxas de cruzamento e

mutação implicam em aumento da velocidade de execução do algoritmo.

A comprovação da efetividade do uso de algoritmos genéticos para resolver o problema de comunalidade permite expandir o rol de problemas que podem ser resolvidos com o método, como sistemas mais complexos, e modelos idem.

#### REFERÊNCIAS

- [1] L. D. Thomas, "A methodology for commonality analysis, with applications to selected Space Station systems", NASA Technical Memorandum 100364, 1989.
- [2] L. D. Thomas, "Commonality Analysis Using Clustering Methods", *Operations Research*, vol. 39, p. 677- 680, 1991.
- [3] J. H. Ward, "Hierarchical Grouping to Optimize an Objective Function", *Journal of the American Statistical Association*, vol. 58, p. 236-244, 1963.
- [4] E. R. Hruschka, R. J. G. B. Campello, A. A. Freitas, A. C. P. L. F. de Carvalho, "A survey of evolutionary algorithms for clustering", *IEEE Transactions on Systems, Man and Cybernetics - Part C: Applications and Reviews*, vol. 39, n. 2, p. 133-155, 2009.
- [5] M.C.Naldi, A. C. P. L. F. de Carvalho, R. J. G. B. Campello, E. R. Hruschka, "Genetic Clustering for Data Mining", In. *Soft Computing for Knowledge Discovery and Data Mining*, p. 113-132, Springer US, 2008.
- [6] A. Tucker, J. Crampton, S. Swift, "RGFGA: An Efficient Representation and Crossover for Grouping Genetic Algorithms", *Evolutionary Computation*, vol. 13, p. 477-499, 2005.
- [7] J. Du, R. Alhajj, K. Barker, "Genetic algorithms based approach to database vertical partition", *J. Intell. Inf. Syst.*, vol. 26, p. 167-183, 2006.
- [8] E. Korkmaz, "Multi-objective Genetic Algorithms for Grouping Problems", *Applied Intelligence*, 2008.