



18 a 21 de novembro de 2014, Caldas Novas - Goiás

ALGORITMO MULTIOBJETIVO PARA OTIMIZAÇÃO DE PROBLEMAS RESTRITOS APLICADOS A INDÚSTRIA

Érico de Oliveira Costa Zini¹, e-mail: zinierico.mat@gmail.com

Alfredo Bonini Neto², e-mail: bonini@tupa.unesp.br

Enio Garbelini³, e-mail: enio.garb@gmail.com

¹FAM - Faculdade de Mirandópolis - Av. São Paulo, 965 - Mirandópolis / SP - CEP 16800-000

²Universidade Estadual Paulista - UNESP – Campus de Tupã – Departamento de Engenharia de Biossistemas. Av. Domingos da Costa Lopes, 780 - CEP 17602-496

³ UNIFADRA - Faculdades de Dracena – R. Bahia, 332 – Dracena / SP – Brazil – CEP 17900-000 / FAI - Faculdades Adamantinenses Integradas – R. 09 de Julho, 730 – Brazil – CEP 17800-00

Resumo: Hoje em dia, com um mundo cada vez mais competitivo, as indústrias estão sujeitas a uma série de restrições impostas devido a vários fatores, tais como problemas ambientais e financeiros, recursos cada vez mais escassos e caros, custos elevados dos serviços e dos salários entre outros. Por isso, como as restrições em torno das indústrias aumentam dia após dia, este trabalho tem por objetivo apresentar uma metodologia para resolução de problemas de otimização com restrições utilizando uma estratégia de busca multiobjetivo, conhecido como algoritmo genético (AG). Os algoritmos genéticos são técnicas de buscas baseadas na teoria da evolução, combinando a sobrevivência dos mais aptos com a troca de informações de uma forma estruturada, onde um problema do mundo real é modelado através de um conjunto de indivíduos que são soluções potenciais que melhor se ajustam ao ambiente correspondendo a uma solução ótima ou quase ótima para o problema. A metodologia é composta por duas fases. Na primeira fase, todo o esforço do algoritmo está direcionado em encontrar, pelo menos, uma única solução factível, descartando completamente a função objetivo. Na segunda fase, aborda-se o problema como biobjetivo, onde se busca a otimização da função objetivo original e maximizar o cumprimento das restrições. Os resultados obtidos são comparados com testes conhecidos na literatura, mostrando que a metodologia proposta obtém com êxito o resultado esperado.

Palavras-chave: Algoritmo Genético, Algoritmo Multiobjetivo, Função Objetivo, Otimização com Restrições.

1. INTRODUÇÃO

A maioria dos problemas do mundo real apresenta, em sua formulação, restrições de igualdade e desigualdade. Na resolução destes problemas os Algoritmos Evolucionários, em especial os Algoritmos Genéticos, têm sido amplamente utilizados. Estas técnicas, ao serem comparadas com os métodos tradicionais de programação, possuem a vantagem de trabalhar com uma menor quantidade de informações não precisando, por exemplo, calcular gradientes, derivadas, hessianas entre outras. Além disso, estas técnicas são de fácil implementação e são ferramentas de busca global.

Durante os últimos anos, vários algoritmos evolucionários foram propostos para resolver problemas de otimização restrita. Considerando a otimização restrita, pode-se dizer que obter uma solução factível, ou seja, uma solução que seja útil na formulação do problema e satisfaça todas as restrições, prevalece, em determinado estágio do algoritmo, sobre otimizar a função objetivo. Existem problemas bastante complexos que são altamente restritos e, neste caso, encontrar uma única solução factível pode ser considerado uma tarefa extremamente difícil. No entanto, deve-se pelo menos esperar que um critério de factibilidade seja satisfeito para todas as tentativas de buscas pela solução ótima e que uma otimização adequada seja alcançada (Venkatraman; Yen, 2005). Sendo assim, objetivou-se elaborar um algoritmo que, em seu primeiro estágio, visa exclusivamente obter, pelo menos, uma única solução factível. Após alcançar com sucesso o primeiro estágio, o algoritmo transfere-se para um segundo estágio que visa realizar a otimização simultânea da função objetivo e da violação das restrições. Neste estágio um problema mono-objetivo de otimização restrita é tratado como um problema biobjetivo e, desta forma, além de buscar a solução ótima global factível do problema, procura-se também obter soluções que violam de forma aceitável as restrições impostas na tentativa de obter um ganho importante nos valores da função objetivo.

Enfim, o objetivo principal deste trabalho foi elaborar um Algoritmo Genético que visa atender a factibilidade e, posteriormente, a factibilidade e otimalidade simultaneamente. Estas características do algoritmo proposto fazem com que sua estrutura seja eficiente e especializada na resolução de problemas da indústria que possuem variáveis contínuas e que são, preferencialmente, altamente restritos, visto que, obter soluções factíveis para estes tipos de problemas é bastante difícil e, além disso, estes problemas muitas vezes permitem uma pequena margem de violação das restrições impostas para obter uma melhoria importante no valor da função objetivo.

Para verificar o desempenho do algoritmo proposto, utilizam-se 2 problemas testes que são bastante utilizados como referência na literatura especializada. Desta forma, obtêm-se os resultados provenientes da aplicação do algoritmo proposto a estes problemas e, posteriormente, realiza-se uma comparação dos resultados obtidos aqui com as soluções já reportadas na literatura utilizando outros algoritmos.

2. METODOLOGIA

2.1. Algoritmo Genético

O Algoritmo Genético é um ramo dos algoritmos evolucionários e como tal pode ser definido como uma técnica de busca que se baseia no processo da evolução natural.

Analisando invenções como, por exemplo, o avião; podem-se perceber características semelhantes à de um pássaro nesta invenção. Há um bom tempo o homem vem se baseando nas características e princípios da natureza para a criação de máquinas, métodos e técnicas que na maioria das vezes visam melhorar ou facilitar a vida em nosso cotidiano (Castro, 2001).

Dentro destes conceitos, fundamentado no processo de seleção natural proposto por Darwin e nos mecanismos da genética surge a teoria dos Algoritmos Genéticos. Este algoritmo foi inventado por Holland na década de 70. Holland estudou formalmente a evolução das espécies e propôs um modelo computacional heurístico que, quando implementado, poderia oferecer soluções de boa qualidade para problemas extremamente difíceis que não podiam ser resolvidos computacionalmente até aquela época (Linden, 2006).

Os Algoritmos Genéticos são técnicas inspiradas na teoria de Darwin, ou seja, são técnicas de buscas baseadas na teoria da evolução, combinando a sobrevivência dos mais aptos com a troca de informações de uma forma estruturada, onde um problema do mundo real é modelado através de um conjunto de indivíduos que são soluções potenciais que melhor se ajustam ao ambiente.

De forma análoga com a teoria da evolução, esses indivíduos são selecionados, se reproduzem e sofrem mutação, obtendo deste modo uma nova geração de indivíduos que também atendem as necessidades do ambiente. Após certo número de gerações espera-se convergir para uma geração de elite que corresponda a uma solução ótima ou quase ótima para o problema.

Desde então, os Algoritmos Genéticos começaram a se expandir por toda a comunidade científica, gerando uma série de aplicações que puderam ajudar a resolver problemas extremamente importantes. Uma grande aplicação dos Algoritmos Genéticos está em problemas de busca, onde dado um conjunto de elementos ou indivíduos, deseja-se encontrar aquele ou aqueles que melhor atendam a certas condições especificadas (Castro, 2001).

Para implementar um algoritmo genético, devem-se realizar os seguintes passos (Pozo et al., 2005):

- Escolhe-se adequadamente uma representação para os indivíduos da população. Geralmente, esta representação é feita utilizando a codificação binária, pois a aplicação dos operadores genéticos de recombinação e mutação são mais simples de serem empregadas. Posteriormente, escolhe-se uma população inicial, normalmente formada por indivíduos criados aleatoriamente;
- Avalia-se adequadamente toda população segundo algum critério determinado por uma função que mede a qualidade do indivíduo (função de aptidão ou “fitness”). Desta forma, os melhores indivíduos são aqueles que apresentam função de aptidão de melhor qualidade;
- Estabelece-se uma estratégia de seleção dos indivíduos como base para criação de um novo conjunto de indivíduos (nova população);
- Estabelece-se um mecanismo que permita implementar os operadores genéticos de recombinação e mutação. A nova população é obtida aplicando sobre os indivíduos selecionados os referidos operadores;
- Repetem-se os passos acima até que: um indivíduo de qualidade aceitável seja encontrado, um número preestabelecido de passos seja atingido ou o algoritmo não consiga mais mostrar evolução, ou seja, não se consegue melhorar a incumbente já encontrada;

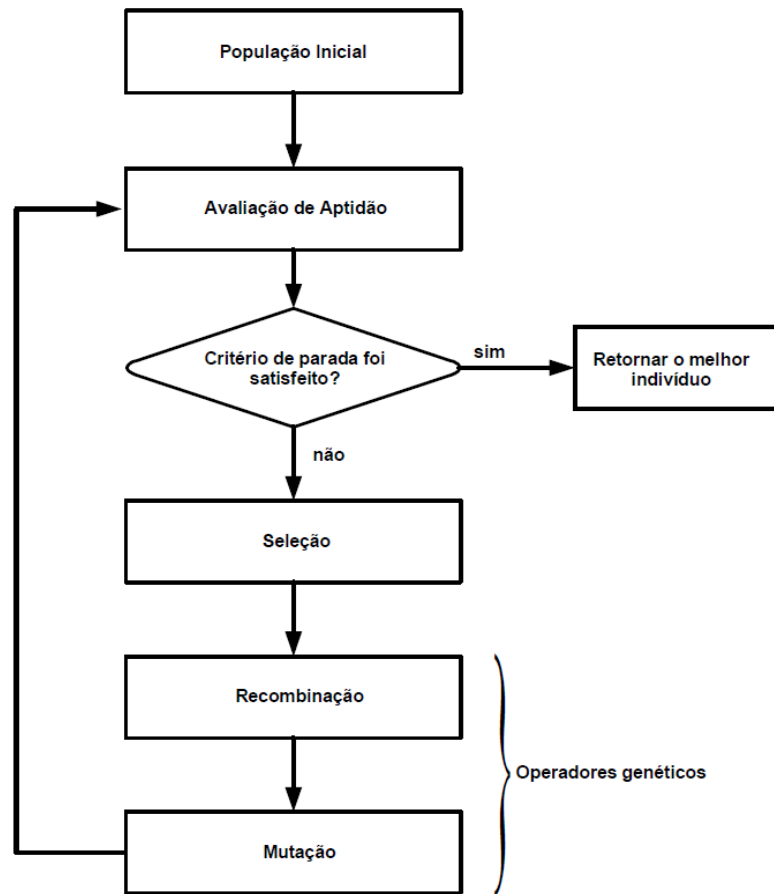


Figura 1. Estrutura básica de um algoritmo genético

2.2. Abordagem Proposta

O algoritmo proposto é composto de duas fases. Na primeira fase, a função objetivo original é completamente desconsiderada e o problema de otimização concentra-se somente em minimizar o grau de violação das restrições. Na execução desta fase, utiliza-se um escalar de violação das restrições, e, com isso, atribuir seu valor de aptidão, define-se a violação do indivíduo na j -ésima restrição como (Venkatraman; Yen, 2005):

$$c_j(\vec{x}) = \begin{cases} \max(0, g_j(\vec{x})) & \text{para } j = 1, \dots, q \\ \max(0, |h_j(\vec{x}) - \delta|) & \text{para } j = q + 1, \dots, m \end{cases} \quad (1)$$

sendo que $||$ denota o operador de valor absoluto.

Para dar a todas as restrições o mesmo grau de importância, cada violação de restrição individual $c_j(\vec{x})$ calculada em (1) são normalizadas.

Primeiramente, encontra-se a violação máxima de cada restrição na população usando a seguinte expressão:

$$c_{\max}(j) = \max_{\vec{x}} c_j(\vec{x}) \quad (2)$$

Estes valores de máxima violação de cada restrição em toda a população são utilizados para normalizar cada violação calculada em (1). Finalmente, as violações de restrições normalizadas de cada indivíduo da população são somadas e divididas pela quantidade de restrições m , produzindo um escalar de violação para cada um destes indivíduos. Obviamente, estes escalares assumirão valores entre 0 e 1. Cada escalar é representado por $v(\vec{x})$ e calculado pela seguinte expressão:

$$v(\vec{x}) = \frac{\sum_{j=1}^m \frac{c_j(\vec{x})}{c_{\max}(j)}}{m} \quad (3)$$

Através da expressão (3) calcula-se um único parâmetro que representa o grau de violação normalizada que abrange todas as restrições para cada indivíduo da população. Este escalar é denominado *escalar de violação de restrição*.

Na segunda fase a violação das restrições $v(\vec{x})$ e a função objetivo original $f(\vec{x})$ devem ser minimizadas simultaneamente dentro de um espaço de busca modificado. O indivíduo factível que possui o melhor valor para a função objetivo $f(\vec{x})$ será a incumbente atual do espaço de busca, ou seja, armazena-se a melhor solução factível do problema.

2.3. O Problema de Produção Industrial

Tem-se observado que a maioria das empresas do setor produtivo não utilizam ferramentas de otimização nos seus processos de tomadas de decisões, tornando-se assim muito empírico a elaboração de seus “mix” de produção, devido principalmente ao elevado número de variáveis e restrições envolvidas no processo (Barreto, 2001).

Uma indústria para produzir n produtos P_j necessita de m recursos F_i . Para cada unidade do produto P_j são necessários a_{ij} unidades do recurso F_i . De cada recurso F_i só existe a quantidade b_i ($b_i \geq 0$). Sabendo que cada unidade do produto P_j fornece um lucro c_j , então, qual a quantidade x_j que deve ser produzida de cada produto P_j para que o lucro seja o maior possível?

Sejam x_1, \dots, x_n as quantidades produzidas dos produtos P_1, \dots, P_n respectivamente. Como para cada recurso F_i tem-se uma restrição do tipo b_i ($b_i \geq 0$), então:

$$v(\vec{x}) = \begin{cases} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n \leq b_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n \leq b_2 \\ \cdot \quad \quad \cdot \quad \quad \cdot \quad \quad \cdot \\ \cdot \quad \quad \cdot \quad \quad \cdot \quad \quad \cdot \\ \cdot \quad \quad \cdot \quad \quad \cdot \quad \quad \cdot \\ a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n \leq b_m \end{cases}$$

Como não se pode produzir uma quantidade negativa de produtos P_j , tem-se que:

$$x_1, x_2, x_3, \dots, x_n \geq 0$$

O interesse é determinar o ponto $x^* = (x_1^*, x_2^*, \dots, x_n^*)$ que satisfaça as m restrições de recursos F_i e as n restrições de positividade das quantidades a serem produzidas, e ao mesmo tempo maximize a função objetivo:

$$f(x_1, x_2, \dots, x_n) = c_1x_1 + c_2x_2 + \dots + c_nx_n$$

O modelo do problema é, portanto:

$$\left\{ \begin{array}{l} \text{Max } f(x_1, x_2, \dots, x_n) = c_1x_1 + c_2x_2 + \dots + c_nx_n \\ \text{s.a.} \\ \begin{array}{l} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n \leq b_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n \leq b_2 \\ \cdot \quad \quad \cdot \quad \quad \cdot \quad \quad \cdot \\ \cdot \quad \quad \cdot \quad \quad \cdot \quad \quad \cdot \\ \cdot \quad \quad \cdot \quad \quad \cdot \quad \quad \cdot \\ a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n \leq b_m \end{array} \end{array} \right. = \left\{ \begin{array}{l} \text{Max } f(\vec{x}) \\ \text{s.a.} \\ v(\vec{x}) \end{array} \right.$$

ou

$$\left\{ \begin{array}{l} \text{Max } f(x_1, x_2, \dots, x_n) = \sum_{j=1}^n c_j x_j \\ \text{s.a.} \\ \sum_{j=1}^n a_{ij} x_j, \text{ com } i \in \{1, \dots, m\} \\ x_j \geq 0, \text{ com } j \in \{1, \dots, n\} \end{array} \right.$$

3. RESULTADOS

O primeiro problema a ser resolvido é:

$$\text{Minimizar } G1(\vec{x}) = 5x_1 + 5x_2 + 5x_3 + 5x_4 - 5\sum_{i=1}^4 x_i^2 - \sum_{i=5}^{13} x_i$$

s.a.

$$2x_1 + 2x_2 + x_{10} + x_{11} \leq 10$$

$$-8x_1 + x_{10} \leq 0$$

$$-2x_4 - x_5 + x_{10} \leq 0$$

$$2x_1 + 2x_3 + x_{10} + x_{12} \leq 10$$

$$-8x_2 + x_{11} \leq 0$$

$$-2x_6 - 2x_7 + x_{11} \leq 0$$

$$2x_2 + 2x_3 + x_{11} + x_{12} \leq 10$$

$$-8x_3 + x_{12} \leq 0$$

$$-2x_8 - x_9 + x_{12} \leq 0$$

e as variáveis limitadas por:

$$0 \leq x_i \leq 1, \quad i = 1, \dots, 9$$

$$0 \leq x_i \leq 100, \quad i = 10, 11, 12$$

$$0 \leq x_{13} \leq 1$$

O problema possui 13 variáveis e 9 restrições lineares e, neste caso, não foi possível obter sua representação gráfica. A função G1 é quadrática com seu mínimo global em:

$$\vec{x}^* = (1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 3, 3, 1)$$

sendo $G1(\vec{x}^*) = -15$. Seis das nove restrições estão ativas no ótimo global, ou seja, todas estão ativas exceto as três seguintes:

$$-8x_1 + x_{10} \leq 0, \quad -8x_2 + x_{11} \leq 0 \quad \text{e} \quad -8x_3 + x_{12} \leq 0$$

O segundo problema é:

$$\text{Maximizar } G2(\vec{x}) = \frac{\left| \sum_{i=1}^n \cos^4(x_i) - 2 \prod_{i=1}^n \cos^2(x_i) \right|}{\sqrt{\sum_{i=1}^n i x_i^2}}$$

s.a.

$$\prod_{i=1}^n x_i \geq 0,75$$

$$\sum_{i=1}^n x_i \leq 7,5n$$

e as variáveis limitadas por:

$$0 \leq x_i \leq 10, \quad \text{para } 1 \leq i \leq n$$

A função G2 é não-linear e seu máximo global é desconhecido. No entanto, para $n = 20$ soluções \vec{x}^* no qual $G2(\vec{x}^*) = 0,803553$ foram encontradas. A seguir ilustram-se alguns gráficos deste problema para $n = 2$. Primeiramente, mostra-se o gráfico Fig. 2 da função objetivo $G2(\vec{x})$ que segue:

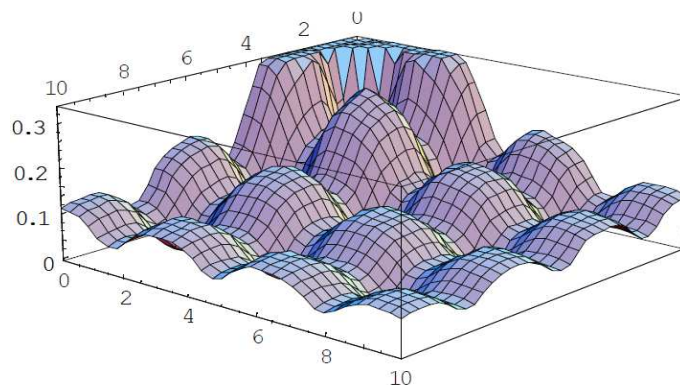


Figura 2. Gráfico da função $G2(\bar{x})$ para $n = 2$

Analisando a Fig. 2, percebe-se que a função $G2(\bar{x})$ possui vários picos, sendo assim uma função é multimodal. O gráfico foi representado com as variáveis de decisão x_1, x_2 variando no intervalo de $[0, 10]$. Na Fig. 3 que segue representam-se no mesmo eixo as curvas de nível das restrições $g_1(x_1, x_2)$ e $g_2(x_1, x_2)$ do problema G2, ou seja, $g_1(\bar{x}) = 0$ e $g_2(\bar{x}) = 0$.

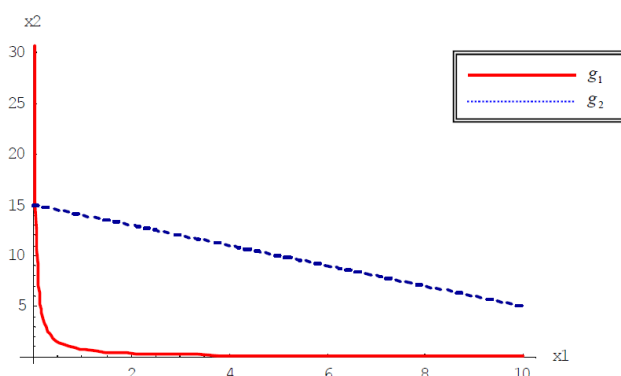


Figura 3. Gráfico das curvas de nível de $g_1(x_1, x_2)$ e $g_2(x_1, x_2)$ para o problema G2

A região factível ocupa uma grande parte do espaço de busca e é dada pela região entre os gráficos de g_1 e g_2 . Obviamente, deve ser respeitado o intervalo $[0, 10]$ para as variáveis x_1 e x_2 .

A taxa de recombinação (ρ_r) e a taxa de mutação (ρ_m) utilizadas foram iguais para todos os casos testes, ou seja, $\rho_r = 90\%$ e $\rho_m = 1\%$. O tamanho da população (n_p) para todos os casos também foi a mesma, ou seja, $n_p = 15$ indivíduos. O número de iterações (gerações) máximo foram de 5000 e foram executadas 30 simulações para cada caso. Para o problema teste G2, utilizou-se $k = 20$ (número de variáveis de decisão). Na primeira fase onde é aplicada a *Diminuição Progressiva da Tolerância de Aceitação das Restrições Complexas (DPTARC)* utilizou-se $\varepsilon_{\max} = 0,4$ e $\varepsilon_{\min} = 0,0001$, e para a estratégia de redução da margem de violação $\varepsilon = (1 - \tau)\varepsilon$ utilizou-se $\tau = 0,5$.

No algoritmo proposto, não foi utilizado nenhuma codificação com cadeias binárias, ou seja, utilizou a codificação dos indivíduos com parâmetros reais. Portanto, isto implicou na mudança dos operadores genéticos de recombinação e mutação.

A seguir, apresenta-se a Tab. 1 reproduzida de (Venkatraman; Yen, 2005), (Koziel; Michalewicz, 1999). Nesta tabela são citados para cada um dos problemas características tais como: objetivo do problema, número de variáveis (n), tipo da função objetivo (f), porcentagem de factibilidade em relação ao espaço de busca (ρ), número de desigualdades lineares (DL), número de igualdades não-lineares (EN), número de desigualdades não-lineares (DN) e quantidade de restrições ativas no ponto ótimo (a).

Tabela 1. Algumas características dos 11 problemas testes.

Problema	Objetivo	n	Tipo de f	ρ	DL	EN	DN	a
G1	Minimizar	13	Quadrática	0,0111%	9	0	0	6
G2	Maximizar	k	Não linear	99,8474%	0	0	2	1

A porcentagem de factibilidade em relação ao espaço de busca (ρ) é dada por $\rho = |F \cap S|/|S|$, e foi determinada experimentalmente gerando 1000000 de soluções aleatoriamente no espaço de busca S e verificando se estas pertenciam a região factível F . Para o problema G2 foi considerado $k = 50$ (número de variáveis de decisão) para a realização deste experimento (Koziel; Michalewicz, 1999). No algoritmo proposto, como já foi dito, utilizou-se $k = 20$ (número de variáveis de decisão).

Analisando a Tab. 1 é possível perceber claramente a variedade de características dos 2 problemas testes utilizados. Estas variedades estão relacionadas com os objetivos dos problemas (maximização e minimização), com os diferentes tipos de função objetivo (quadrática e não-linear) e os diferentes tipos de restrições (desigualdades não-lineares e lineares).

Para produzir os resultados utilizando o algoritmo proposto realizou-se 30 simulações para cada caso. Uma das primeiras observações que se pode fazer em relação a este experimento é que em todas as tentativas de busca pela solução ótima foram encontradas soluções factíveis. A obtenção de soluções factíveis em todas as simulações é creditada à fase I do algoritmo proposto que utiliza a *Diminuição Progressiva da Tolerância de Aceitação das Restrições Complexas (DPTARC)* que busca, exclusivamente, a satisfação de restrições. A referida fase, que utilizou a *DPTARC*, não aumenta significativamente o esforço computacional do algoritmo proposto, visto que, foram realizados testes para problema altamente restritos e verificou-se que o número médio de gerações para obtenção da solução factível foi bastante pequeno.

No problema G2, a solução factível é sempre encontrada na população inicial gerada aleatoriamente. Isto acontece pelo fato das regiões factíveis deste problema, ocuparem uma grande parte do espaço de busca. De fato, pois como se pode verificar na Tab. 1, a porcentagem de factibilidade em relação ao espaço de busca (ρ) para este problema é dada, respectivamente, por 99,8474%.

O problema G1 possui nove restrições lineares e, conseqüentemente, a região factível é convexa. Assim, não houve dificuldades para minimizar a distância representada pelos escalares de violação das restrições e obter soluções factíveis. Especificamente, o número médio de gerações para encontrar a primeira solução factível foi de 11,24.

As Tab. 2 e Tab. 3 a seguir apresentam-se os melhores resultados (valores ótimos e variáveis de decisão) alcançados na segunda fase, ou seja, apresentam-se a melhor solução encontrada em todas as 30 simulações para cada problema.

Tabela 2. Melhor solução encontrada para o Problema G1.

<i>PROBLEMA G1</i>		
Valor Objetivo		-14,99990
Variáveis de Decisão	x_1	1,00000
	x_2	1,00000
	x_3	1,00000
	x_4	1,00000
	x_5	1,00000
	x_6	1,00000
	x_7	1,00000
	x_8	1,00000
	x_9	1,00000
	x_{10}	3,00000
	x_{11}	3,00000
	x_{12}	2,99990
	x_{13}	1,00000

Tabela 3. Melhor solução encontrada para o Problema G2.

<i>PROBLEMA G2</i>		
Valor Objetivo		0,80318350
Variáveis de Decisão	x_1	3,15616
	x_2	3,11157
	x_3	3,09893
	x_4	3,07220
	x_5	3,05592
	x_6	3,03985
	x_7	2,95829
	x_8	2,91929
	x_9	0,48135
	x_{10}	0,48921
	x_{11}	0,48045
	x_{12}	0,48541
	x_{13}	0,45520
	x_{14}	0,47170
	x_{15}	0,46356
	x_{16}	0,45162
	x_{17}	0,44490
	x_{18}	0,44345
	x_{19}	0,44910
	x_{20}	0,45714

Com estes melhores resultados alcançados pelo algoritmo proposto para cada um dos problemas é feito um comparativo com os valores já alcançados por outros autores aplicando outras estratégias nestes mesmos 2 problemas. A Tab. 4 a seguir mostra esta comparação.

Tabela 4. Comparação dos melhores resultados.

<i>Problema</i>	<i>Valor Ótimo</i>	<i>Koziel e Michalewicz (1999)</i>	<i>Runarsson e Yao (2000)</i>	<i>Farmani e Wright (2003)</i>	<i>Venkatraman e Yen (2005)</i>	Algoritmo Proposto
Min. G1	-15	-14,7864	-15,0000	-15,0000	-14,9999	-14,99990
Max. G2	0,803553	0,799530	0,803515	0,802970	0,803190	0,80318350

Analisando a última coluna da Tab. 4, pode-se perceber que o algoritmo proposto teve um desempenho satisfatório, pois as melhores soluções obtidas estão bem próximas dos valores ótimos apresentados na segunda coluna da tabela.

Para o problema G1, o algoritmo proposto encontrou o valor -14,99990 que está muito próximo do ótimo global -15. Para o problema G2, o resultado foi bastante satisfatório, sendo que, obteve-se o valor 0,80318350 que está próximo do ótimo conhecido 0,803553.

4. CONCLUSÕES

Neste trabalho, foi proposto um Algoritmo Genético composto de duas fases. A qualidade deste algoritmo foi validada através de 2 problemas testes encontrados na literatura especializada. Foi feito um comparativo dos resultados alcançados e verificou-se que os valores obtidos pelo algoritmo proposto são bastante próximos aos alcançados na literatura.

O algoritmo proposto apresentou a vantagem de não utilizar parâmetros que dependem de informações subjetivas do problema e, deste modo, mostrou-se bastante genérico, pois mesmo os problemas testes possuindo características bem diversificadas, os parâmetros de controle utilizados foram os mesmos para todos estes problemas. A proposta apresentada pode ser abordada em problemas de diferentes campos da indústria e, preferencialmente, em problemas que possuem variáveis contínuas e são altamente restritos. Enfim, conclui-se que o algoritmo proposto apresentou um bom desempenho, visto que, foram obtidos valores ótimos próximos aos valores ótimos já reportados na literatura especializada.

5. AGRADECIMENTOS

Os autores agradecem o apoio do CNPq e da Fundunesp.

REFERÊNCIAS

- Barreto, A. L. de Oliveira. 2001. *Otimização em Processos Multidivisionais Aplicados à Industria Têxtil*. R. Cient. Fac. Lour. Filho - v.1, n.1.
- Castro, R. E. 2001. *Otimização de estruturas, com multiobjetivos via algoritmo genético de pareto*. 202f. Tese (Doutorado) - Coordenação dos Programas de Pós Graduação de Engenharia - COPPE, Universidade Federal do Rio de Janeiro - UFRJ, Rio de Janeiro, 2001.
- Farmani, R.; Wright, J. 2003. *Self-adaptive fitness formulation for constrained optimization*, IEEE Transactions on Evolutionary Computation, New York, v.7, n.5, p. 445-455.
- Koziel, S.; Michalewicz Z. 1999. *Evolutionary algorithms, homomorphous mappings, and constrained parameter optimization*. Evolutionary Computation, Cambridge, v.7, p. 19-44.
- Linden, R. *Algoritmos genéticos: uma importante ferramenta da inteligência computacional*. Rio de Janeiro: Brasport, 2006.
- Pozo, A.; Cavalheiro A. F.; Ishida, C.; Spinosa, E.; Rodrigues E. M. 2005. *Computação evolutiva*. Universidade Federal do Paraná, 61p. (Grupo de Pesquisas em Computação Evolutiva, Departamento de Informática-Universidade Federal do Paraná)
- Runarsson, T.; Yao, X. 2000. *Stochastic ranking for constrained evolutionary optimization*. IEEE Transactions on Evolutionary Computation, New York, v.4, p. 344-354.
- Venkatraman, S.; Yen, G. G. 2005. *A Generic framework for constrained optimization using genetic Algorithms*. IEEE Transactions on Evolutionary Computation, New York, v.9, n.4, p. 424-435.

RESPONSABILIDADE AUTORMAL

“Os autores são os únicos responsáveis pelo conteúdo deste trabalho”.

MULTIOBJECTIVE ALGORITHM FOR CONSTRAINED OPTIMIZATION PROBLEMS APPLIED TO INDUSTRY

Érico de Oliveira Costa Zini¹, e-mail: zinierico.mat@gmail.com
Alfredo Bonini Neto², e-mail: bonini@tupa.unesp.br
Enio Garbelini³, e-mail: enio.garb@gmail.com

¹FAM - Faculty Mirandópolis – Av. São Paulo, 965 - Mirandópolis / SP- Brazil - CEP 16800-000

²UNESP - São Paulo State University – Campus Tupã - Biosystems Engineering Department. Av. Domingos da Costa Lopes, 780 – Brazil - CEP 17602-496

³ UNIFADRA - Faculty Dracena – R. Bahia, 332 – Dracena / SP – Brazil – CEP 17900-000 / FAI - Adamantinense Integrated Faculty – R. 09 de Julho, 730 – Brazil – CEP 17800-000

Abstract. Nowadays, with a world increasingly competitive, industries are subject to a number of constraints due to several factors, such as environmental and financial problems, increasingly scarce and expensive resources, high cost of services and wages between others. Therefore, as the constraint around the industries increase day by day, this work aims at presenting a methodology for solving constrained optimization problems using a multi-objective search strategy, known as genetic algorithm (GA). Genetic algorithms are search techniques based on the theory of evolution by combining the survival of the fittest with the exchange of information in a structured way, where a real-world problem is modeled through a set of individuals who are potential solutions that best adjust to the environment corresponding to an optimal or nearly optimal solution to the problem. The methodology consists of two stages. In the first phase, all effort of the algorithm is directed at finding at least one feasible solution, completely discarding the objective function. In the second phase, directs the problem as bi-objective, where intend to optimize the original objective function and maximize the compliance with the constraint. The results obtained are compared with tests known in the literature, showing that the proposed method successfully obtains the desired result.

Keywords: Genetic algorithms, Multiobjective algorithm, Objective function, Constrained optimization.