Investigando a Utilização de Abordagens Compactas nas Estratégias de Evolução

Anderson T. Sergio, Sidartha Carvalho, Marco Antônio Q. Costa Rego
CIn, Centro de Informática
UFPE, Universidade Federal de Pernambuco
Recife, Brasil
ats3@cin.ufpe.br, salc@cin.ufpe.br, maqct@cin.ufpe.br

***Resumo*—Algoritmos evolucionários compactos têm se mostrado uma eficiente alternativa para resolução de problemas de otimização em ambientes computacionais com pouco poder de processamento. Nesse tipo de solução, uma distribuição de probabilidade simula o comportamento de uma população, buscando assim economia de memória. Diversos algoritmos compactos foram propostos, dentre eles o algoritmo genético compacto e a evolução diferencial compacta. O trabalho em questão busca investigar a utilização das abordagens compactas em outro algoritmo evolucionário importante: estratégias de evolução. Este artigo propõe duas versões compactas para diferentes abordagens das estratégias de evolução. Experimentos foram realizados e os resultados analisados. Os resultados mostraram que, a depender do tipo do problema, utilizar a versão compacta das estratégias de evolução pode ser compensador.**

Palavras-Chave—sistemas adaptativos, algoritmos evolucionários compactos, estratégias de evolução, algoritmos de distribuição estimativa.

***Abstract*— Compact evolutionary algorithms have proven to be an efficient alternative for solving optimization problems in computing environments with a few processing power. In this kind of solution, a probability distribution simulates the behavior of a population, thus searching memory economy. Several compact algorithms have been proposed, including the compact genetic algorithm and compact differential evolution. This work aims to investigate the use of compact approaches in other important evolutionary algorithm: evolution strategies. This article proposes two different approaches for compact versions of evolution strategies. Experiments were performed and the results analyzed. The results showed that, depending on the nature of problem, to use the compact version of Evolution Strategies can be rewarding.**

Keywords—adaptive systems, compact evolutionary algorithms, evolution strategies, estimation of distribution algorithms.

I. INTRODUÇÃO

Em muitas aplicações do mundo real, um problema de otimização deve ser resolvido mesmo em situações em que um grande poder computacional não esteja disponível devido às limitações de custo e/ou de espaço. Esta situação pode ocorrer em problemas de robótica e de controle de processos. Para contornar essas adversidades, uma solução proposta foi o desenvolvimento dos chamados algoritmos evolucionários compactos (cEAs, do inglês *Compact Evolutionary Algorithms*).

Os algoritmos evolucionários compactos pertencem à categoria dos algoritmos de distribuição de estimativa (EDAs, do inglês *Estimation of Distribution Algorithms*) [1]. Nessa classe de algoritmos, uma população não é armazenada e processada. Para dar continuidade ao processo de otimização, faz-se o uso de uma representação estática dos indivíduos. Essa característica permite um menor número de parâmetros armazenados em memória, fazendo com que sua execução requeira menos espaço de armazenamento quando comparado aos algoritmos evolucionários padrão.

Diversos cEAs foram propostos, dentre eles o algoritmo genético compacto (cGA, do inglês *Compact Genetic Algorithm*) [2] e a evolução diferencial compacta (cDE, do inglês *Compact Differential Evolution*) [3]. Embora as estratégias de evolução (ES, do inglês *Evolution Strategies*) sejam soluções bastante utilizadas, nenhuma versão compacta foi proposta para estes algoritmos. Este estudo propõe duas versões de estratégias de evolução compactas, a c(1+1)-ES e a c(μ, λ)-ES. O objetivo principal do trabalho é investigar a utilização das abordagens compactas nesse algoritmo clássico, através da análise de resultados obtidos em bases de dados de *benchmark*.

O presente trabalho está organizado da seguinte maneira: uma descrição geral dos algoritmos compactos e das estratégias de evolução usados como base para as abordagens propostas neste trabalho está na sessão II. A sessão III descreve os algoritmos c(1+1)-ES e c(μ, λ)-ES e discute seus princípios de funcionamento e detalhes. A Sessão IV mostra os resultados numéricos e é subdividida em duas partes, de acordo com as abordagens propostas. As conclusões e sugestões de trabalhos futuros encontram-se na sessão V.

II. BACKGROUND

A. Algoritmos Compactos

Algoritmos compactos são estimativas de algoritmos de distribuição que imitam o comportamento de base populacional por meio de uma representação probabilística da população de soluções candidatas. Estes algoritmos têm um comportamento semelhante no que diz respeito aos algoritmos baseados na população, mas exigem uma memória muito menor [3].

O primeiro cEA proposto foi o algoritmo genético compacto, introduzido em [2]. O cGA simula o comportamento de um algoritmo genético (GA, do inglês *Genetic Algorithm*) com codificação binária [4]. Em [2] pode ser visto que cGA tem um desempenho quase tão bom quanto o do GA. Como esperado, a principal vantagem de um cGA quando comparado a um GA padrão é a economia de memória.

No cGA original, um vetor binário de tamanho *n* é gerado aleatoriamente atribuindo 0,5 de probabilidade para cada gene receber os valores 0 ou 1. Este vetor que descreve as probabilidades, inicializado com *n* valores iguais a 0,5, é denominado vetor de probabilidade (PV, do inglês *Probability Vector*). Através do PV, dois indivíduos são amostrados e os seus valores de aptidão são calculados. A solução vencedora, isto é, aquela caracterizada por um maior desempenho, altera o PV com base no parâmetro N_p, chamado de população virtual. Se a solução vencedora em seu gene de posição *i* possui valor 1, enquanto a solução perdedora possui 0 na mesma posição, o valor de probabilidade na posição *i* do PV é aumentado em uma quantidade $\frac{1}{Np}$. Caso contrário, o PV é reduzido em uma quantidade $\frac{1}{Np}$ na posição em questão. Se os genes na posição *i* exibirem o mesmo valor tanto para o vencedor quanto para o perdedor, a probabilidade do PV na posição *i* não é modificada.

A versão na qual esse estudo se baseia, o algoritmo genético compacto de valores reais (rcGA, do inglês *Real-Valued Compact Genetic Algotithm*), foi introduzida em [5]. O rcGA é um algoritmo compacto inspirado pelo cGA que exporta a lógica compacta para um domínio de valores reais, obtendo-se assim um algoritmo de otimização com um elevado desempenho, apesar da quantidade limitada de memória. No rcGA o PV não é um vetor, mas uma matriz $n \times 2$:

$$PV^t = [\mu^t, \sigma^t] \quad (1)$$

onde μ e σ são, respectivamente, os vetores que contém, para cada variável de projeto, valores de média e desvio padrão de uma função de distribuição de probabilidade de Gauss truncada dentro do intervalo [-1,1]. No início do processo de otimização, para cada variável *i*, $\mu^1[i] = 0$ e $\sigma^1[i] = \lambda$, onde λ é uma grande constante positiva (λ = 10). A inicialização dos valores *σ[i]* é feita com o objetivo de simular uma distribuição uniforme. Subsequentemente, um indivíduo é amostrado como elite. Um novo indivíduo é gerado e comparado com a elite. Assim como no cGA, no rcGA a solução vitoriosa também altera o bias do PV. A regra de atualização para cada elemento de μ é dada por:

$$\mu^{t+1}[i] = \mu^t[i] + \frac{1}{N_p}(vencedor[i] - perdedor[i]) \quad (2)$$

onde N_p é o tamanho da população. A regra de atualização de σ é dada por:

$$(\sigma^{t+1}[i])^2 = (\sigma^t[i])^2 + (\mu^t[i])^2 - (\mu^{t+1}[i])^2 + \frac{1}{N_p}(vencedor[i]^2 - perdedor[i]^2) \quad (3)$$

μ é a média do PV, σ é o desvio-padrão do PV, vencedor é o passo de mutação que gerou o melhor indivíduo e perdedor é o passo de mutação que gerou o pior indivíduo.

Outras versões compactas de algoritmos genéticos podem ser encontradas na literatura. Em [6], foi proposto o algoritmo genético compacto estendido (ecGA, do inglês *Extended Compact Genetic Algorithm*). No ecGA, a distribuição de probabilidades utilizada é uma classe de modelos de probabilidades conhecidos como modelos de produto marginal. Uma versão híbrida do ecGA com o algoritmo de Nelder-Mead pode ser vista em [7] e o estudo de sua escalabilidade é mostrado em [8]. Adicionalmente, uma variante memética foi apresentada em [9], com o objetivo de aumentar a convergência do algoritmo na presença de um grande número de dimensões. As diversas versões compactas dos algoritmos genéticos têm sido utilizadas em aplicações práticas implementadas em *hardware*, onde os recursos computacionais podem ser menores ([10] [11] [12]).

Com ideias similares aos algoritmos genéticos compactos, [3] propõe uma versão compacta da evolução diferencial, o cDE. Na evolução diferencial, um conjunto de vetores de parâmetros é gerado aleatoriamente no início do processo, cobrindo todo o espaço de busca. O algoritmo então recombina esses vetores visando a minimização ou maximização de uma função objetivo. A versão compacta da evolução diferencial foi aplicada a um caso de estudo relacionado ao treinamento online de uma rede neural implementada diretamente em um microcontrolador. Os resultados numéricos e experimentais comprovaram a eficiência do algoritmo proposto.

B. Estratégias de Evolução

Estratégias de evolução são uma subclasse de algoritmos de busca direta baseados na natureza e pertencentes à classe de algoritmos evolutivos [13]. ES usam mutação, recombinação e seleção aplicadas a uma população de indivíduos contendo soluções candidatas de forma a encontrar melhores soluções iterativamente. O algoritmo foi inicialmente proposto em 1974 [14].

Um algoritmo evolucionário é geralmente descrito de acordo com características como representação dos indivíduos, tipo da recombinação, tipo da mutação e seleção dos pais. A seguir, cada uma dessas características será discutida.

ES são tipicamente utilizados para problemas de otimização contínua. A representação padrão dos indivíduos é dado por um vetor de reais $x_1, ..., x_n$, onde cada x_i representa uma variável de ponto flutuante. Entretanto, os algoritmos mais modernos de ES utilizam esse vetor apenas como uma parte do genótipo. Os indivíduos podem conter alguns parâmetros de estratégia, que influenciam diretamente a operação de mutação. Assim, o genótipo pode ser completamente definido por $(x_1, ..., x_{n,,} \sigma_1, ..., \sigma_{n\sigma,} \alpha_1, ..., \alpha_{n\alpha})$.

A operação de mutação é baseada numa distribuição normal que requer os parâmetros média e desvio-padrão. De uma maneira geral, o mecanismo utiliza um valor para adicionar um ruído formado a partir dessa distribuição. Esse valor, denominado tamanho do passo de mutação, faz parte do genótipo, que pode evoluir. A operação de mutação pode ser realizada de diversas maneiras. Abaixo, a descrição dos mecanismos utilizados neste trabalho.

Na mutação não correlacionada com um passo, um único valor por indivíduo é calculado, multiplicando esse valor por uma distribuição lognormal. Abaixo, as equações de atualização:

$$\sigma' = \sigma * \exp\big(\tau * N(0,1)\big) \quad (4)$$

$$x_i' = x_i + \sigma' * N(0,1) \quad (5)$$

σ é o passo de mutação, x_i um gene do genótipo e τ uma taxa de aprendizado, geralmente proporcional a $1/n^{1/2}$. O passo de mutação é o mesmo em cada direção.

Já na mutação não correlacionada com passo n, existe um passo de mutação para cada gene do indivíduo. As equações são as que seguem:

$$\sigma_i' = \sigma_i * \exp(\tau' * N(0,1) + \tau * N_i(0,1)) \quad (6)$$

$$x_i' = x_i + \sigma_i' * N_i(0,1) \quad (7)$$

Nesse caso, τ' é a taxa de aprendizado global, a mesma para o indivíduo. τ é taxa de aprendizado específica, permitindo mutações em diferente direções. A primeira é proporcional a $1/(2n)^{1/2}$ e a segunda proporcional a $1/(2n^{1/2})^{1/2}$.

Na recombinação das estratégias de evolução, dois pais geram um filho. Atuando por posição, a recombinação pode ser intermediária ou discreta. Na recombinação intermediária, os filhos são formados pela média dos valores dos pais. Já na recombinação discreta, o filho é gerado selecionando-se o gene de um dos pais, a partir de uma distribuição de probabilidades. Os parâmetros de estratégia (mutação) são geralmente recombinados de maneira intermediária, enquanto que os parâmetros objetivo (genes) são recombinados de forma discreta [13]. A recombinação também difere no que diz respeito aos pais utilizados: usando dois pais previamente selecionados para gerar um filho ou selecionando dois pais diferentes para cada posição.

Os pais são selecionados através de uma distribuição aleatória uniforme. Assim, cada indivíduo tem a mesma probabilidade de ser selecionado, independente de sua aptidão.

As versões mais clássicas de ES coexistem com versões mais modernas. Um desses algoritmos mais recentes é o CMA-ES (do inglês *Covariance Matrix Adaptation Evolution Strategy*) [15]. Nesse algoritmo, as dependências entre as variáveis na distribuição de probabilidades são representados por uma matriz de covariância. O CMA-ES leva essa matriz em consideração quando realiza as operações de adaptação.

III. Estratégia de Evolução Compacta

A seguir serão apresentadas as duas versões propostas para as estratégias de evolução compactas. De um modo geral, elas diferem na representação da população.

A. Algoritmo c(1+1)-ES

Desde que as estratégias de evolução foram inicialmente propostas, diversas abordagens foram investigadas, variando-se parâmetros como o tipo da mutação e a organização da população. Outra importante variação é o mecanismo de seleção dos sobreviventes. De uma forma geral, esses diferentes mecanismos são conhecidos como (μ, λ) e (μ + λ). Sendo μ o tamanho da população em determinada iteração (os pais) e λ o tamanho da prole, o primeiro mecanismo indica que a população da próxima iteração não levará em conta os pais, sendo, portanto, uma abordagem não elitista. No segundo caso, a população da próxima iteração será composta tanto pelos pais quanto pela prole.

Na abordagem conhecida como (1+1)-ES, a população é formada por apenas dois indivíduos, que vão sendo adaptados ao longo da execução do algoritmo. Propor uma versão compacta para essa abordagem não proporciona economia de memória, portanto. Mas o c(1+1)-ES (a versão compacta do (1+1)-ES) foi proposto para investigar a utilização das abordagens compactas em um algoritmo de estratégias de evolução mais simples, verificando se seu desempenho é adequado.

O algoritmo proposto é executado seguindo a sequencia de passos da figura 1.

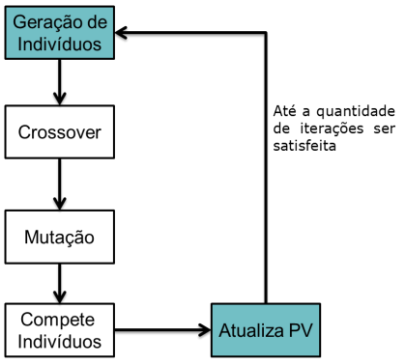

Figura 1. Algoritmo do c(1+1)-ES.

Diferentemente da estratégia convencional, a (1+1)-ES, a inicialização dos indivíduos dessa abordagem utiliza um vetor de probabilidades. Esse PV armazena a média e o desvio padrão que é utilizado para gerar os valores dos genes dos indivíduos. Essa abordagem de uso do PV induz que os indivíduos sejam gerados em uma área promissora do espaço de busca. Ao final de cada geração, o PV é atualizado de acordo com as equações descritas no algoritmo rcGA (equações 2 e 3).

No (1+1)-ES convencional, os indivíduos são gerados aleatoriamente por uma distribuição normal de média 0 e desvio padrão 1, deixando totalmente ao acaso o valor do gene. Comparando essas abordagens, percebe-se que a versão compacta tem mais chances de gerar indivíduos mais aptos. Com o passar do tempo, a geração dos indivíduos é direcionada para uma área promissora e não avança de maneira aleatória.

As duas abordagens diferem no que foi mencionado acima, na geração dos indivíduos e na utilização do PV. Os outros

operadores como mutação, recombinação e a função de avaliar a aptidão de cada indivíduo são iguais.

B. *Algoritmo c(μ, λ)-ES*

Eiben e Smith [13] indicam que o mecanismo não elitista na seleção dos sobreviventes é mais utilizado e apresenta os melhores resultados. De acordo com esses autores, (μ, λ) deve ter preferência porque é melhor em deixar ótimos locais e mover-se para regiões promissoras, já que descarta todos os pais e não preserva soluções desatualizadas. Adicionalmente, ao utilizar o mecanismo (μ+λ), valores de mutação ruins podem persistir na população por muito tempo.

Tendo em vista essas questões, este trabalho também propõe um algoritmo que investiga a utilização de abordagens compactas no chamado (μ, λ)-ES. Essa abordagem será chamada de c(μ, λ)-ES.

O algoritmo c(μ, λ)-ES inspira-se no rcGA e cDE para conferir ao (μ, λ)-ES um tratamento compacto. A figura 2 mostra o esquema geral de execução do c(μ, λ)-ES. Em seguida, a descrição do algoritmo.

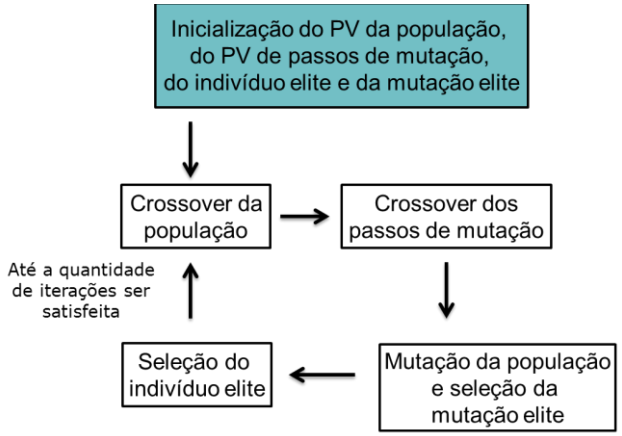

Figura 2. Algoritmo do c(μ, λ)-ES.

Incialmente o vetor de probabilidades PV da população é inicializado, com média e desvio-padrão. A média é gerada a partir de uma distribuição de probabilidades uniforme entre um limite inferior e superior para cada função objetivo, valores estes definidos empiricamente. O desvio-padrão é gerado a partir de uma distribuição normal N(0, 1). Um indivíduo elite também é gerado a partir de uma distribuição uniforme entre os limites inferior e superior da função objetivo. A mesma lógica é aplicada ao PV dos passos de mutação e à mutação elite. A diferença são os limites, que no caso da mutação, é dado por -1 e 1. No c(μ, λ)-ES, a abordagem de mutação é a não correlacionada com *n* passos.

Após a inicialização desses vetores, as iterações são iniciadas, correspondendo ao número de gerações. Em cada geração, o primeiro passo é realizar a recombinação da população. Seguindo as características do rcGA e do cDE, dois pais são amostrados a partir do PV da população. Um indivíduo atual é gerado a partir da combinação desses dois pais. A combinação pode ser intermediária ou discreta. Em seguida, lógica semelhante é utilizada para a geração de um vetor de mutações atual. Dois pais são amostrados a partir do PV de mutações e recombinados de forma intermediária ou discreta.

Nesse ponto da geração, o algoritmo tem em memória um indivíduo atual e um vetor de passos de mutação atual. Seguindo as regras de atualização padrão das estratégias de evolução, esse indivíduo é mutado pelo vetor de mutações atual e pelo vetor de mutações elite. O indivíduo atual é atualizado pelo melhor indivíduo gerado a partir dessas duas mutações, e essa informação é utilizada para atualizar o PV de mutações. A atualização é semelhante ao que ocorre no c(1+1)-ES, dada pelas equações 2 e 3.

Após a mutação do indivíduo atual e atualização do PV de passos de mutação, ocorre uma competição entre esse indivíduo e o indivíduo elite. O resultado dessa competição influencia a atualização do PV de população, novamente de acordo com as equações 2 e 3. Todas essas operações são realizadas até que o número máximo de gerações seja atingido.

IV. RESULTADOS

Para validar os resultados, os algoritmos propostos foram comparados com suas respectivas versões não compactas. O c(1+1)-ES foi comparado com o (1+1)-ES e o c(μ, λ)-ES foi comparado com o (μ, λ)-ES. A comparação foi realizada a partir dos melhores valores de aptidão alcançados pelos algoritmos a partir de 10 funções-problema. A seguir, as funções utilizadas [16], cada uma com dimensão n = 2: Beale, Booth, Dixon, Griewank, Hump, Levy, Matyas, Rastrigin, Rosenbrock, Sphere.

Dando validade estatística às comparações, o teste de Wilcoxon foi aplicado à execução de 30 rodadas, com um grau de confiança de 95%. Nas tabelas que apresentam os resultados dos testes estatísticos, "+" indica o caso em que o algoritmo compacto superou o tradicional em determinada função; um "=" indica que a diferença dos resultados é estatisticamente irrelevante e, portanto, os algoritmos têm a mesma performance; um "-" indica que a abordagem tradicional superou a abordagem compacta.

A. *Algoritmo c(1+1)-ES*

Esta sessão apresenta os resultados alcançados pelo algoritmo c(1+1)-ES, a comparação com o algoritmo original (1+1)-ES e o resultado do teste estatístico para comprovar que os resultados são estatisticamente equivalentes. O algoritmo compacto se mostrou equivalente no quesito desempenho comparado ao (1+1)-ES convencional.

Na mutação dos indivíduos, foi usada uma estratégia adaptada da mutação não correlacionada com passo de mutação 1. Na abordagem compacta, cada gene do indivíduo possui um σ que representa o passo de mutação para aquele gene. A abordagem convencional utiliza um σ para todos os genes do mesmo indivíduo. A seguir os parâmetros de cada algoritmo.

- (1+1)-ES: μ = 1; λ = 1; gerações = 100; recombinação da população = intermediária; mutação = versão adaptada da mutação não correlacionada com um passo de mutação.

- c(1+1)-ES: Uso do PV para geração dos indivíduos μ = 1; λ = 1; gerações = 100; recombinação da população = intermediária; mutação = versão adaptada da mutação não correlacionada com um passo de mutação.

Na tabela I é possível observar a média e os desvios-padrão das melhores aptidões alcançadas pelos indivíduos. Pode-se perceber que os valores se assemelham muito nas duas abordagens.

A tabela II mostra a validação dos resultados pelo teste de Wilcoxon. Pode-se que na maioria das funções os algoritmos são equivalentes e também que na função de Rastrigin o algoritmo compacto se saiu melhor. Já na função de Matyas o algoritmo compacto foi superado pelo seu concorrente.

TABELA I
MÉDIA DAS MELHORES APTIDÕES ALCANÇADAS

Função	(1+1)-ES	C(1+1)-ES
Beale	0.004880	0.009391
	(0.026282)	(0.050574)
Booth	0.003839	0.006861
	(0.020678)	(0.036951)
Dixon	0.029809	0.002496
	(0.160531)	(0.013441)
Griewank	4.4862E-4	0.001237
	(0.002415)	(0.006662)
Hump	0.001659	0.004053
	(0.008939)	(0.021828)
Levy	5.8960E-4	0.001608
	(0.003175)	(0.008660)
Matyas	2.3911E-4	5.0806E-4
	(0.001287)	(0.002736)
Rastrigin	0.040865	0.008293
	(0.220068)	(0.044663)
Rosenbrock	0.001090	0.010790
	(0.005870)	(0.058107)
Sphere	2.2695E-4	0.001001
	(0.001222)	(0.005390)

TABELA II
TESTE DE WILCOXON PARA VALIDAÇÃO

Função	(1+1)-ES vs. c(1+1)-ES
Beale	=
Booth	=
Dixon	=
Griewank	=
Hump	=
Levy	=
Matyas	-
Rastrigin	+
Rosenbrock	=
Sphere	=

Em geral os resultados mostraram-se estatisticamente semelhantes. Confirmando que a primeira abordagem compacta possui depempenho equivalente à versão convencional, a próxima sessão mostra os resultados para a versão compacta de um algoritmo de estratégias de evolução mais sofisticado.

B. Algoritmo c(μ, λ)-ES

Esta sessão apresenta os resultados alcançados pelo c(μ, λ)-ES, bem como uma comparação com a versão tradicional do algoritmo. Para versão tradicional do algoritmo, dois conjuntos de parâmetros foram testados. Os parâmetros foram selecionados ou seguindo indicações de trabalhos anteriores ou empiricamente. A seguir, o conjunto de parâmetros utilizados.

- (μ, λ)-ES-1: μ = 10; λ = 10; gerações = 100; recombinação da população = discreta; recombinação dos passos de mutação = intermediária; limite inferior para passo de mutação = -1; limite superior para passo de mutação = 1.
- (μ, λ)-ES-2: μ = 10; λ = 20; gerações = 100; recombinação da população = discreta; recombinação dos passos de mutação = intermediária; limite inferior para passo de mutação = -1; limite superior para passo de mutação = 1.
- c(μ, λ)-ES: gerações = 100; recombinação da população = discreta; recombinação dos passos de mutação = intermediária; limite inferior para passo de mutação = -1; limite superior para passo de mutação = 1.

A tabela III mostra a média das melhores aptidões alcançadas por cada um dos algoritmos em cada uma das funções testadas, em 30 execuções. Entre parêntesis, os valores de desvio-padrão. A tabela IV mostra a comparação estatística entre o (μ, λ)-ES-1 e o c(μ, λ)-ES e, de forma similar, a comparação entre o (μ, λ)-ES-2 e o c(μ, λ)-ES.

Através da análise das tabelas, é possível observar que a abordagem compacta proposta teve desempenho superior ao ES tradicional com o primeiro conjunto de parâmetros, μ = 10 e λ = 10, em metade das funções de teste. Na outra metade, a diferença foi estatisticamente irrelevante. Além do ganho na melhor aptidão encontrada, deve-se salientar o ganho esperado devido às características de um algoritmo compacto. A economia de memória é evidente, visto que no c(μ, λ)-ES, a população é armazenada em um vetor n x 2, sendo n a dimensão do problema.

Entretanto, é sabido e foi constatado empiricamente que o desempenho da abordagem (μ, λ)-ES pode melhorar. A ideia mais direta para o ganho em termos da aptidão alcançada é aumentar o tamanho da população. Eiben e Smith [13] indicam que as abordagens (μ, λ) das estratégias de evolução melhoram significativamente quando $\lambda > \mu$. Essa é justamente a característica da abordagem (μ, λ)-ES-2, onde λ = 20 e μ = 10. Como esperado, o desempenho melhorou significativamente e o c(μ, λ)-ES não conseguiu superar o ES tradicional com esse conjunto de parâmetros em nenhuma das funções testadas.

Levando em conta os resultados alcançados nas funções testadas, é possível inferir que o desempenho da abordagem não compacta tende a ser melhor do que a abordagem compacta na medida em que é executada com parâmetros adequados. Esse comportamento é esperado, tendo em vista que as abordagens compactas inclinam-se a perder desempenho devido a utilização do vetor de probabilidades em detrimento ao armazenamento em memória de uma população.

A vantagem da abordagem compacta, como constatado empiricamente, é a economia de memória. Mas mesmo quando comparado com a abordagem (μ, λ)-ES-1, que utiliza uma população de tamanho 10, o desempenho do c(μ, λ)-ES foi superior.

TABELA III
MÉDIA DAS MELHORES APTIDÕES ALCANÇADAS

Função	(μ, λ)-ES-1	(μ, λ)-ES-2	c(μ, λ)-ES
Beale	0.105397	0.065293	0.685472
	(0.216403)	(0.218540)	(1.703286)
Booth	3.873424	0.000698	0.029533
	(5.991175)	(0.000756)	(0.023305)
Dixon	0.286664	0.000639	0.017207
	(0.525051)	(0.000761)	(0.019740)
Griewank	0.461445	0.255029	0.660719
	(0.265004)	(0.264967)	(0.668455)
Hump	0.218676	0.000641	0.052358
	(0.377594)	(0.000499)	(0.145693)
Levy	0.007013	0.005408	0.114306
	(0.010092)	(0.018034)	(0.430628)
Matyas	0.092499	0.000033	0.001468
	(0.162598)	(0.000048)	(0.002028)
Rastrigin	2.310114	0.175241	2.321291
	(1.650039)	(0.186175)	(1.428543)
Rosenbrock	0.849357	0.033554	1.174561
	(1.187724)	(0.066951)	(2.063353)
Sphere	0.063853	0.000170	0.007260
	(0.086762)	(0.000170)	(0.007272)

TABELA IV
TESTE DE WILCOXON PARA VALIDAÇÃO

Função	(μ, λ)-ES-1 vs. c(μ, λ)-ES	(μ, λ)-ES-2 VS. c(μ, λ)-ES
Beale	=	-
Booth	+	-
Dixon	+	-
Griewank	=	-
Hump	+	-
Levy	=	-
Matyas	+	-
Rastrigin	=	-
Rosenbrock	=	-
Sphere	+	-

V. CONCLUSÕES E TRABALHOS FUTUROS

Este artigo introduziu o conceito de estratégia de evolução compacta e propôs duas variantes de algoritmos, o c(1+1)-ES e o c(μ, λ)-ES. Ambas as variantes não requerem hardware potente, a fim de exibir um alto desempenho. Pelo contrário, os algoritmos propostos fazem uso de uma quantidade limitada de memória, a fim de realizar a otimização.

O algoritmo c(1+1)-ES provou ser equivalente ao (1+1)-ES original, validando, assim, a utilização de um vetor de probabilidades. A versão c(μ, λ)-ES, no entanto, mostrou que sua performance é inferior quando o (μ, λ)-ES utiliza um valor de λ alto, ou seja, gera uma grande quantidade de cromossomos durante a reprodução e ocupa uma grande quantidade de memória. Todavia, nas ocasiões em que a versão compacta tem desempenho inferior, há uma grande economia de memória, mostrando a utilidade do algoritmo compacto quando existem poucos recursos computacionais e a solução encontrada não precisa ser a melhor.

Um possível trabalho futuro seria a implementação da estratégia de evolução com mutação correlacionada e sua comparação com os demais algoritmos já propostos neste artigo. Uma comparação com outros trabalhos e algoritmos da literatura também deve ser levada em consideração. Um outro ponto importante seria a aplicação desse algoritmo em um problema real de implementação em *hardware*, fazendo valer sua vantagem de economia de memória.

REFERÊNCIAS

[1] P. Larrañaga, J. A. Lozano, "Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation," Boston, MA: Kluwer, 2001.

[2] G. R. Harik, F. G. Lobo, D. E. Goldberg, "The compact genetic algorithm," IEEE Trans. Evol. Comput., vol. 3, no. 4, pp. 32–53, Nov. 1999.

[3] E. Mininno, F. Neri, F. Cupertino, D. Naso, "Compact Differential Evolution," IEEE Trans. Evol. Comput., vol. 15, no. 1, pp. 287–297, Fev. 2011.

[4] H. J. Holland, "Adaptation in Natural and Artificial Systems," University of Michigan Press, 1975.

[5] E. Mininno, F. Cupertino, D. Naso, "Real-valued compact genetic algorithms for embedded microcontroller optimization," IEEE Trans. Evol. Comput., vol. 12, no. 2, pp. 203–219, Apr. 2008.

[6] K. Sastry, D. E. Goldberg, "On extended compact genetic algorithm." Univ. Illinois at Urbana-Champaign, Urbana, Tech. Rep. 2 000 026, 2000.

[7] K. Sastry, G. Xiao, "Cluster optimization using extended compact genetic algorithm," Univ. Illinois at Urbana-Champaign, Urbana, Tech. Rep. 2 001 016, 2001.

[8] K. Sastry, D. E. Goldberg, D. D. Johnson, "Scalability of a hybrid extended compact genetic algorithm for ground state optimization of clusters," Mater. Manuf. Processes, vol. 22, no. 5, pp. 570–576, 2007.

[9] R. Baraglia, J. I. Hidalgo R. Perego, "A hybrid heuristic for the traveling salesman problem," IEEE Trans. Evol. Comput., vol. 5, no. 6,pp. 613–622, Dec. 2001.

[10] C. Aporntewan, P. Chongstitvatana, "A hardware implementation of the compact genetic algorithm," in Proc. IEEE Congr. Evol. Comput., vol. 1. 2001, pp. 624–629.

[11] C. Aporntewan , P. Chongstitvatana, "A hardware implementation ofthe compact genetic algorithm," in Proc. IEEE Congr. Evol. Comput., vol. 1. 2001, pp. 624–629.

[12] Y. Jewajinda, P. Chongstitvatana, "Cellular compact geneticalgorithm for evolvable hardware," in Proc. Int. Conf. Electr.Eng./Electron. Comput. Telecommun. Inform. Technol., vol. 1. 2008,pp. 1–4.

[13] Eiben, A. E., Smith, J. E., 2003. Introduction to Evolutionary Computing. Springer, Berlin/Heidelberg.

[14] Hans-Paul Schwefel (1974): Numerische Optimierung von Computer-Modellen (PhD thesis). Reprinted by Birkhäuser (1977).

[15] Hansen, N. "The CMA evolution strategy: a comparing review", Towards a new evolutionary computation. Advances on estimation of distribution algorithms, Springer, 2006. pp. 1769–1776

[16] HEDAR, Abdel-rahman. Global optimization test problems. 2013. Disponível em: <http://www-optima.amp.i.kyoto-u.ac.jp/member/student/hedar/Hedar_files/TestGO_files/Page364.htm>. Acesso em: 09 dez. 2013.