Algoritmo de otimização metaheurística inspirado na dinâmica de um jogo de basquete

Gylles Ricardo Ströher, Gisely Luzia Ströher

Resumo: Um novo algoritmo de otimização metaheurística inspirado nos eventos de uma partida de basquete, chamado de *Basketball Game Optimization Algorithm* (BGOA), para otimização multidimensional global é proposto. Os principais eventos de uma partida de basquete foram modelados matematicamente provendo operadores simples, eficientes e de fácil implementação computacional. O BGOA foi testado com algumas funções *standard benchmark* com e sem restrições e os resultados obtidos confirmaram a validade do algoritmo.
Palavras chave: Otimização, Máximo e Mínimo Absolutos.

Metaheuristic optimization algorithm inspired by the dynamics of a basketball game

Abstract: A new metaheuristic optimization algorithm inspired by the events of a basketball match, called the Basketball Game Optimization Algorithm (BGOA), for global multidimensional optimization is adopted. The main basketball starting events were mathematically modeled, providing simple operators and easy computational implementation. BGOA has been tested with standard and unrestricted benchmark functions and the results have been confirmed as validated by BGOA.
Key-words: Global Optimization, Basketball Game Optimization Algorithm

1. Introdução

Existem diversos algoritmos de otimização combinatória inspirados em eventos ou comportamentos da natureza, destes se destacam o Simulated Annealing (METROPOLIS e et al., 1953) que foi inspirado em uma analogia com o recozimento, um fenômeno termodinâmico. Os Algoritmos Genéticos introduzidos pelo trabalho de Holland (HOLLAND, 1962), os quais realizam procedimentos de busca no espaço de soluções viáveis, utilizando regras probabilísticas que simulam o processo biológico da evolução natural.

Mais recentemente, inspirado no padrão de agrupamento de pássaros e peixes, surgiu o Particle Swarm Algoritm, (KENNEDY e EBERHART, 1995) que consiste basicamente em mimetizar o comportamento adotado em multidões de indivíduos. Paralelamente surgiram técnicas inspiradas na capacidade de atuação em conjunto de insetos, surgindo o Ant Colony (GAMBARDELLA e DORIGO, 1996), que mimetiza o comportamento real de colônias de formigas.

Todos estes meta-algoritmos brevemente comentados já foram extensamente aplicados com sucesso em diversos estudos disponíveis na literatura, existindo ainda muitos outros com meta heurísticas inspiradas em analogias com fenômenos físicos, químicos, sociais, semânticos ou mesmo em propriedades matemáticas (GOLDBARG et al. 2016). Em termos de implementação computacional, um meta-algoritmo pode ser mais complexo que outro e também o entendimento do(s) eventos(s) que os inspiraram pode muitas vezes não ser de fácil interpretação fenomenológica.

Neste contexto, o presente trabalho tem como objetivo principal apresentar o Basketball Game Optimization Algorithm (BGOA), uma nova abordagem para otimização combinatória inspirado na observação da dinâmica de uma partida real de basquete. Os eventos mimetizados são de forte conhecimento a maioria das pessoas e com operadores simples e de fácil implementação computacional. O artigo é dedicado a apresentar brevemente o BGOA e o seu desempenho para casos de otimização multidimensional com ou sem restrição.

2. Algoritmo BGOA

A invenção do jogo de basquete como é conhecido hoje é atribuída a James Naismith, que em 1891, nos Estados Unidos, estabeleceu 13 regras para o novo jogo. Rapidamente o basquete se tornou um dos esportes mais populares do mundo, tornando-se um esporte olímpico em 1936 nos jogos de Berlin.

Basicamente, o objetivo do jogo é introduzir a bola no cesto da equipe adversária (marcando pontos) e, simultaneamente, evitar que esta seja introduzida no próprio cesto, respeitando as regras do jogo. A equipe que obtiver mais pontos no fim do jogo vence.

Resumidamente, o BGOA baseou-se nos principais eventos da dinâmica de uma partida de basquete, mimetizando sete eventos: (1) posição inicial dos jogadores na quadra, (2) distância de cada jogador do cesto, (3) o jogador que lança a bola em direção ao cesto, (4) posição da bola após arremesso, (5) rebotes, (6) substituições de jogadores e (7) troca das posições dos jogadores em função da localização da bola.

No jogo de basquete o número total de jogadores é 10, entretanto, no BGOA, assim como em outros algoritmos de otimização, o número de elementos de um vetor de entrada, no

caso do BGOA o número de jogadores, depende fortemente do número de dimensões da função objetivo a ser otimizada, não se restringindo ao número formal de jogadores, o mesmo ocorre ao número de substituições de jogadores ou do tempo, além disso, o tempo máximo de 24 segundos para executar uma jogada pode ser maior ou menor, dependendo do número de rebotes.

Em palavras o BGOA é sumarizado no Quadro 1, neste o algoritmo é apresentado em duas colunas. Na primeira coluna o algoritmo é apresentado em forma de palavras e na segunda coluna no formato matemático.

Passo 1: Gerar aleatoriamente a posição de cada jogador.	Passo 1: Gerar X_i
Passo 2: Calcular a distância de cada jogador ao cesto e também obter a posição do jogador mais próximo do cesto.	Passo 2: $f_i = f(X_i)$ $f_{best} = f_{máx.} = f(X_{Best})$
Passo 3: Definir qual dos jogadores será o lançador da bola para o cesto, os jogadores mais próximos do cesto terão maior probabilidade de ser o jogador lançador.	Passo 3: $X_L = X(L)$
Passo 4: Calcular a posição da bola após o lançamento feito pelo jogador definido no passo anterior. Baseado na posição do lançador e talvez na posição média dos jogadores.	Passo 4: $X_b = (C1*X_L + C2*X_{Best})/(C1+C2) \pm C3*rand*X_{Best}$
Passo 5: Se a posição da bola for pior que a posição do lançador fazer n rebotes até que uma melhor posição seja obtida ou o número de rebotes máximo tenha sido atingido.	Passo 5: $f_b = f(X_b)$ $f_L = f(X_L)$ Enquanto $f_b \leq f_L$ ou rebotes ≤ rebotesMax $X_b = X_L + 1*C4*rand*(X_b - X_{Best})$
Passo 6: Calcular as novas posições dos jogadores em função da localização da bola.	Passo 6: $X = X + C5*rand*(X_B - X)$
Passo 7: Substituir o pior jogador pelo melhor jogador e/ou outras substituições.	Passo 7: $f_{worst} = f(X_{worst}) = f_{MIN.}$ $X_{worst} = X_{best}$
Passo 8: Condicionar as posições dos jogadores aos limites da quadra.	Passo 8: LiminteInferior ≤ x < ≤ LimiteSuperior
Passo 9: Se o critério de parada não foi satisfeito, voltar ao passo 2.	Passo 9: Se ite ≥ itemax, fim Senão Passo 2

Quadro 1: Resumo do BMA

Inicialmente, no Passo 1, é gerado aleatoriamente a posição dos jogadores nos limites da quadra de basquete, sendo que o limite da quadra corresponde a faixa dos valores de cada variável independente da função objetivo. Neste passo, opcionalmente, também pode-se distribuir uniformemente os jogadores no perímetro da quadra.

No Passo 2, a distância de cada jogador do cesto é obtida artificialmente por meio das avaliações da função objetivo, no caso, quanto menor o valor da função objetivo para um determinado jogador X_i mais longe este jogador está do cesto adversário, consequentemente, a probabilidade dele ser o jogador escolhido para arremessar a bola em direção ao cesto é menor. O jogador melhor posicionado, X_{best}, é aquele que está mais próximo do cesto, ou seja, com a maior a avaliação da função objetivo. Alternativamente, se

a busca é por um mínimo global pode-se alternar este passo, ou seja, quanto maior o valor da função objetivo para um determinado jogador X_i mais longe este jogador está do cesto adversário.

No passo 3 é escolhido o jogador que deverá arremessar a bola em direção ao cesto. Os jogadores que estiverem posicionados mais perto do cesto possivelmente serão os jogadores que terão mais chances de acertar o cesto, assim as posições X_i dos jogadores com maiores avaliação da função objetivo terão maiores probabilidades de ser o jogador escolhido para arremessar a bola, o jogador escolhido é denominado de jogador lançador, X_L. Como ocorre em uma partida real de basquete, nem sempre o jogador mais perto do cesto é o jogador que arremessa a bola (X_L=X_{best}), pois na dinâmica real de uma partida, por muitas vezes o jogador melhor posicionado (X_{best}) está marcado por um adversário, não está num ângulo favorável para o arremesso, apesar de estar perto da cesta, ou ainda, pode haver um ou mais jogadores que se precipitem, ou ainda algum jogador opte pelo lance de três pontos.

No passo 4, o jogador escolhido, X_L , para lançar a bola a arremessa e esta adquire uma nova posição em quadra. A posição da bola é artificialmente calculada a partir da indicada no passo 4, que é função da média ponderada das posições do melhor jogador (X_{Best}) e do jogador lançador (X_L), pois é esperado que a bola seja lançada em uma direção relativamente perto da posição do jogador melhor posicionado, sendo C1 e C2 constantes que representam a influência do jogador lançador e do jogador melhor posicionado, respectivamente. E também a soma ou a subtração de um produto da constante C3 por um fator aleatório no intervalo 0 a 1. Esta constate C3 representa a qualidade do arremesso pelo jogador lançador, pois nem sempre a bola arremessada entra no cesto.

Passo 5, naturalmente, após o arremesso da bola, podem ocorrer três situações: a bola cai no cesto pontuando a favor do time ou a bola pode cair em uma posição da quadra mais distante (inferior) ou mais próxima ainda àquela que a mesma foi lançada, neste caso pode se ter um ou uma sequencia de rebotes, isto é, novos arremessos até que a bola caia no cesto ou não. Assim, artificialmente, são gerados *n* números de rebotes definidos pelo usuário, por meio da equação indicada no Passo 5, estas novas posições da bola são obtidas, sendo função da posição do lançador e de um termo de quão rápido a bola vai em direção potencial ao jogador de melhor posição, a constante C4 representa a velocidade.

Após definida a posição da bola, novas posições para os jogadores são artificialmente geradas no Passo 6, numa partida de basquete, tipicamente os jogadores se movimentam em direção a bola, assim as posições dos jogadores são calculadas artificialmente por meio da equação indicado no Passo 6 em que a constante C5 representa velocidade média dos jogadores.

Subsequentemente, após a sequencia de eventos já ocorridos, o técnico avaliando o desempenho e o condicionamento dos jogadores pode optar por realizar algumas substituições, assim no Passo 7 é realizadas as substituições, por exemplo, o jogador pior posicionado é substituído pelo jogador melhor posicionado, $X_{worst} = X_{Best}$.

No Passo 8, caso seja geradas posições dos jogadores fora dos limites da quadra, isto é, alguns vetores *X* estejam fora da faixa numérica delimitada pelo usuário, os mesmos são condicionados para os limites da faixa numérica.

Finalmente, caso o critério de parada tenha sido atingido, encerra-se o algoritmo, caso

contrário, inicia-se uma nova iteração a partir do Passo 2. Os marcos principais do BGOA são sumarizados também na Fig. 1 na forma de fluxograma.

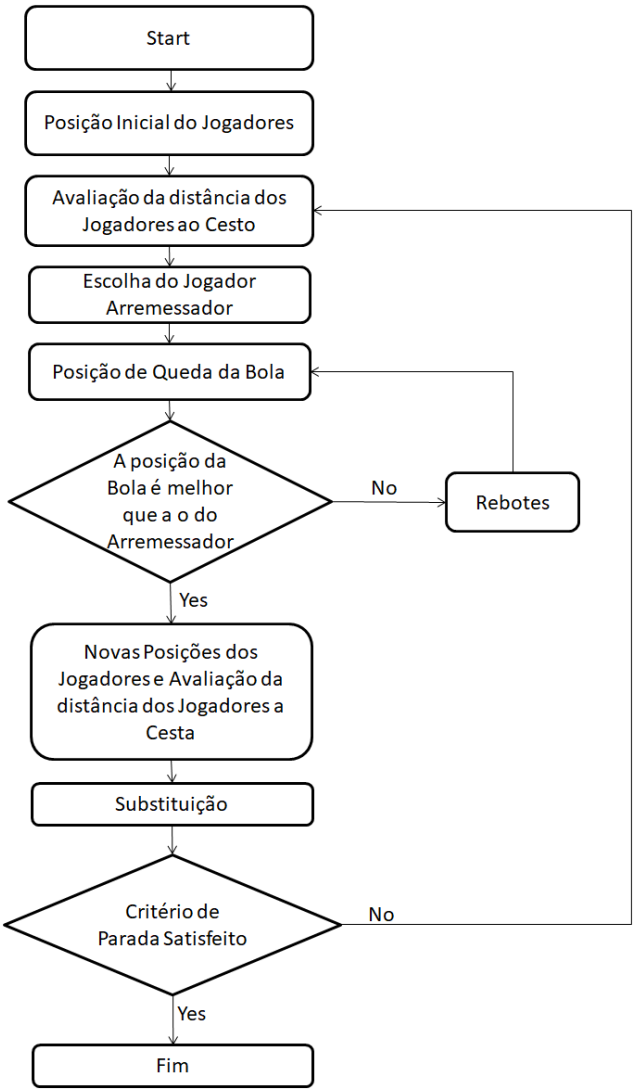


Figura 1: Algoritmo BGOA

3. Resultados e Discussão

Para uma avaliação preliminar do BGOA algumas funções testes para otimização foram utilizadas, em comum as funções apresentam comportamento multimodal. A Tab. 1 sumaria as funções utilizadas.

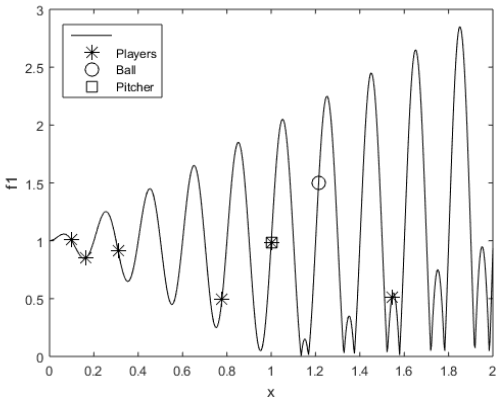
Função	Máximo ou Mínimo
$f_1(x) = \lvert x \sin(10\pi x) + 1 \rvert,\ 0 \le x \le 2$	$f_{1max}(1.851486)=2.85121$
$f_2(x,y)=\lvert x \sin(4x)+1.1\ y\ \sin(2y) \rvert$	$f_{2max}(9.8238,10)=19.8618$

$f_3(x,y,z) = 10(x-1)^2+20(y-2)^2+30*(z-3)^2$	$f_{3min}(0.4377,1.4569,3.1054)=9.3941$
$x+y+z \leq 5$	
$x^2+2y \leq z$	

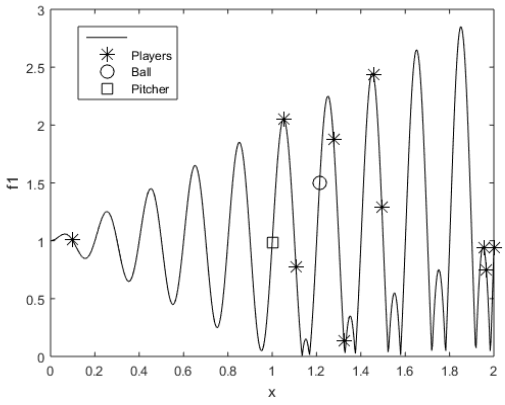
Tabela 1 – Funções utilizadas para avaliação do BGOA

As Figuras 2 a 4 apresentam os resultados obtidos para as funções f1 a f3, nas Figs. 2,3,4 (a) são mostradas as posições dos jogadores obtidas no Passo 1, a posição do jogador que arremessou a bola obtida por meio do Passo 3 e a posição da bola após o arremesso obtido artificialmente pelos Passos 4 e 5. As Figs. 2,3,4 (b) mostram a nova posição dos jogadores após o arremesso da bola, como era esperado os jogadores se deslocam em direção a bola, Passo 6.

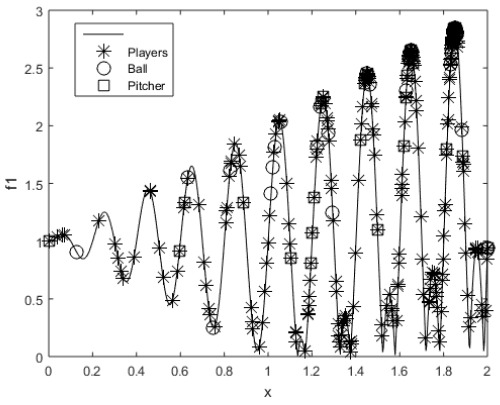
Na Fig. 2,3 (c) são sumarizadas as posições dos jogadores, do arremessador e da bola para várias iterações, observa-se que ao longo do processo iterativo de busca se concentra em torno do ponto ótimo, enquanto a Fig. 3(c) são mostradas as posições dos jogadores após certo número de iterações. As Figs. 2, 3, 4(d) apresentam o comportamento geral do algoritmo BGOA, nestas são dados os históricos das médias das avaliações das funções e do valor f_{max} ou f_{mim}.



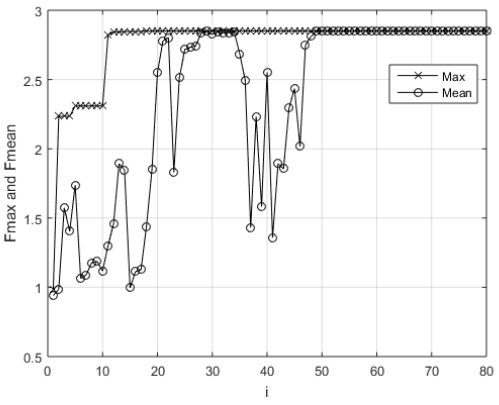
(a)



(b)

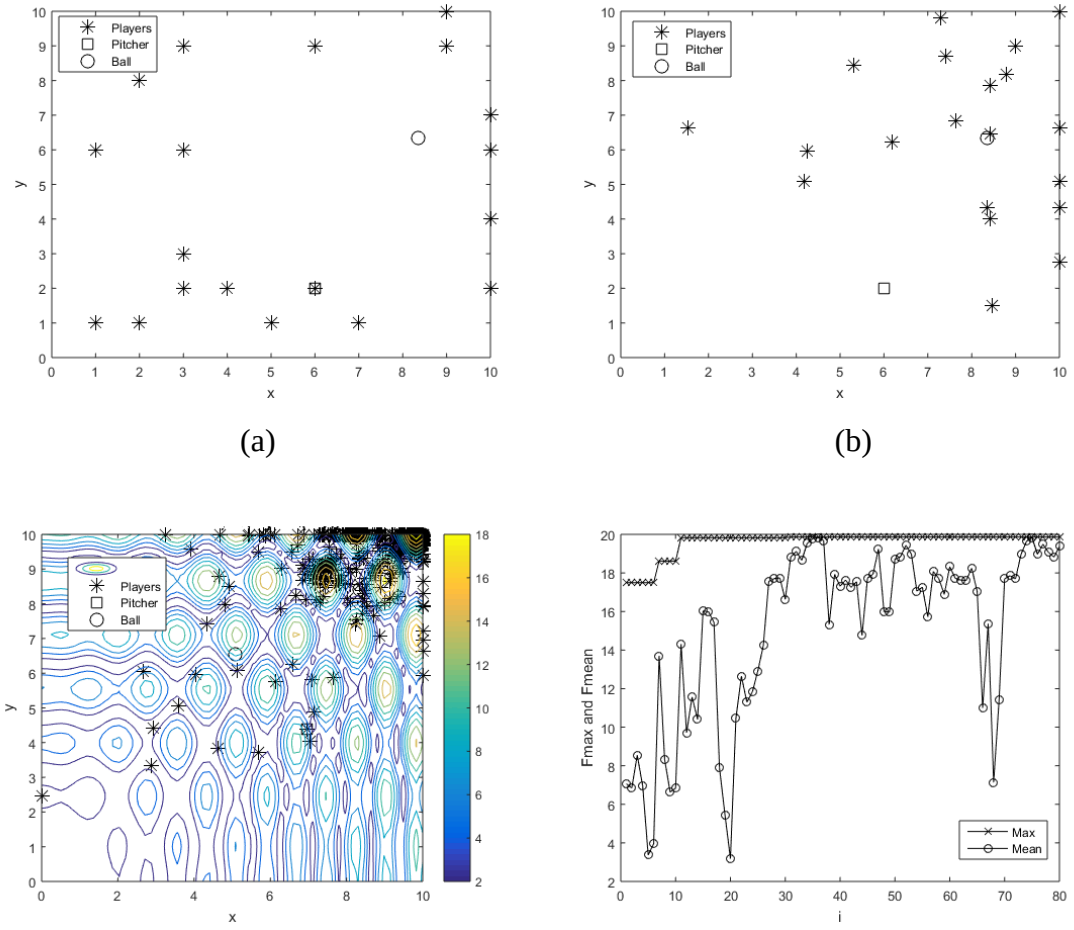


(c)



(d)

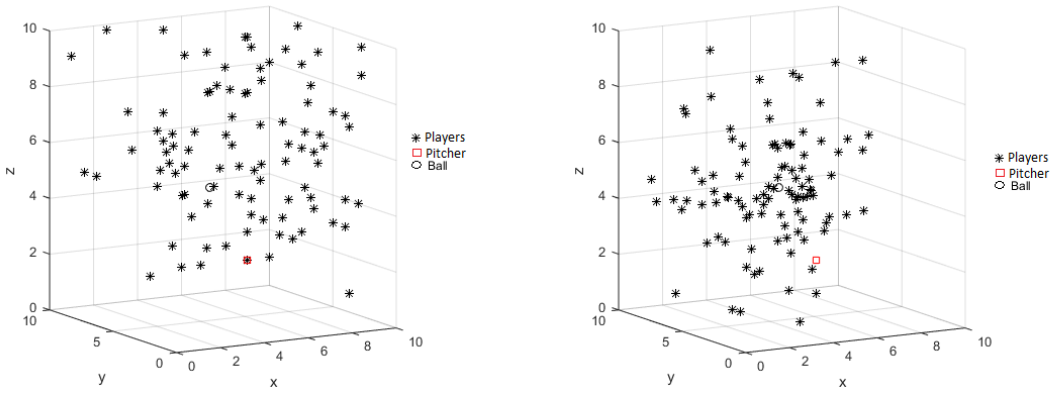
Figura 2: Busca realizada pelo BGOA para a função f 1 . A linha continua representa função f 1 (a) Passos 1, 2,3 e 4. (b) Passos 6, 7 e 8. (c) n iterações do BGOA (d) evolução do valor máximo e médio com o número de iterações



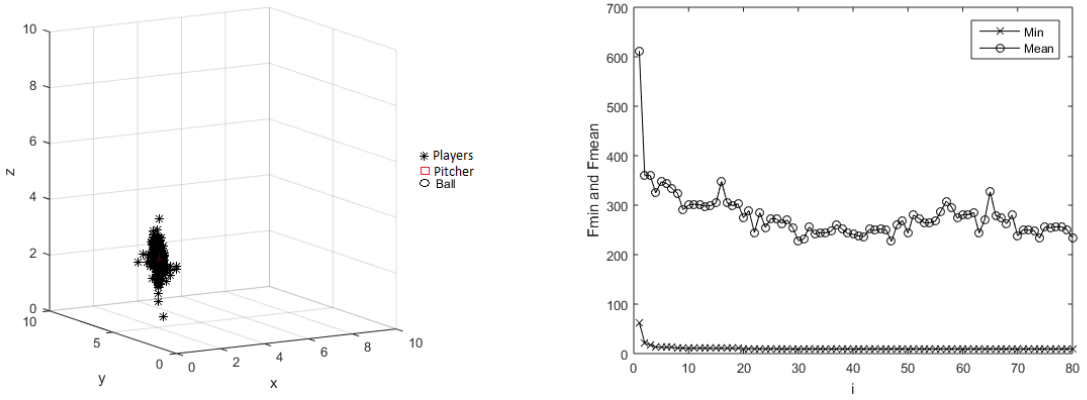
(a) (b)

(c) (d)

Figura 3: Busca realizada pelo BGOA para a função f 2 . A linha continua representa função f 1 (a) Passos 1, 2,3 e 4. (b) Passos 6, 7 e 8. (c) n iterações do BGOA (d) evolução do valor máximo e médio com o número de iterações



(a) (b)



(c) (d)

Figura 4: Busca realizada pelo BGOA para a função f3. A linha continua representa função f 1 (a) Passos 1, 2,3 e 4. (b) Passos 6, 7 e 8. (c) n iterações do BGOA (d) evolução do valor máximo e médio com o número de iterações

As Figs 2, 3 e 4 revelam que o BGOA nos três casos determinou corretamente os pontos de máximo absoluto das funções *f1* e *f2* e o mínimo absoluto para a função *f3*. Sendo necessárias menos de 20 iterações, como BGOA possui operações matemáticas com a geração de números randômicos os perfis mostrados nas Figuras 2 a 4 sofrem variações para cada execução o que naturalmente altera o número de iterações para a convergência da busca.

O BGOA mostrou-se uma meta-heuristica eficaz, particularmente, as Figs 2, 3 e 4(c) mostram que ao longo do processo iterativo as posições dos jogadores se concentram em torno da posição de melhor avaliação da função objetivo seja para o ponto de máximo ou de mínimo global.

Especificadamente para a função *f1*, observa-se na Fig.2 (a) que o jogador que lançou a bola teve um alto valor de avaliação de *f1*, ou seja, fisicamente, dentre os jogadores ele está bem próximo da posição do cesto. Após o arremesso a bola caiu em uma posição melhor que dos outros jogadores, como mostra a Fig. 2(a), o que era esperado com a operação de rebotes do Passo 5. Subsequentemente, os jogadores assumiram posições melhores ainda quando partem em direção da bola, Fig. 2(b). Ao longo do processo os jogadores se concentram na posição de $f1_{max}$ (1,851486). Essencialmente, os mesmos comentários deste parágrafo podem ser feitos para os resultados obtidos com as funções *f2* e *f3*.

Faz-se importante comentar que neste trabalho os resultados foram obtidos utilizando como número de jogadores o valor de 10 vezes o valor de variáveis independentes das funções testadas e 5 para o número máximo de rebotes para cada ciclo iterativo. As demais constantes, C1 a C5 podem ser ajustadas, bem como o número de jogadores e de rebotes para um caso específico e potencialmente obter um melhor desempenho em termos de tempo de processamento e convergência.

4. Conclusões

Um novo algoritmo metaheusritico, chamado de BGOA, foi proposto com a mimetização de sete eventos de uma partida real de basquete. A partir dos resultados obtidos pode-se concluir que o BGOA é constituído de operações simples, eficientes e de fácil implementação computacional, não envolvendo operações matemáticas complexas. O BGOA mostrou-se uma meta-heurística eficaz para os casos testados, mostrando que ao longo do processo iterativo as posições dos jogadores, ou seja, os pontos testados de procura convergem em torno do ponto ótimo da função objetivo.

Referências

GAMBARDELLA, L. M. ; DORIGO, M. Solving symmetric and asymmetric TSPs by ant colonies. **IEEE** p. 622–627, 1996.

HOLLAND, J. H. Outline for a Logical Theory of Adaptive Systems. **J. ACM**, v. 9, n. 3, p. 297–314, 1962.

KENNEDY, J.; EBERHART, R. Particle Swarm Optimization. **Proceedings** of IEEE International Conference on Neural Networks. IV, p. 1942–1948, 1995.

METROPOLIS, NI.; ROSENBLUTH, A. W.; ROSENBLUTH, M. N.; TELLER, A. H.; TELLER, E. Equation of State Calculations by Fast Computing Machines. **The Journal of Chemical Physics**, v. 21, n. 6, p. 1087–1092, 1953.

GOLDBARG, M.; LUNA, H.; GOLDBARG, Elizabeth. **Otimização combinatória e meta-heurísticas: algoritmos e Aplicações**. 1. ed. Rigo de Janeiro: Editora Campus, 2016. . Acesso em: 26 dez 2017.