Proceeding Series of the Brazilian Society of Applied and Computational Mathematics, v. 6, n. 1, 2018.

Trabalho apresentado no XXXVII CNMAC, S.J. dos Campos - SP, 2017.

Proceeding Series of the Brazilian Society of Computational and Applied Mathematics

Controle baseado em redes neurais artificiais para agentes móveis em formação

Vander Luis de Souza Freitas[1]

Doutorando em Computação Aplicada, Instituto Nacional de Pesquisas Espaciais, São José dos Campos, SP

Marcos Gonçalves Quiles[2]

Instituto de Ciência e Tecnologia (ICT), Universidade Federal de São Paulo, São José dos Campos, SP

Elbert E. N. Macau[3]

Laboratório de Computação e Matemática Aplicada, Instituto Nacional de Pesquisas Espaciais, São José dos Campos, SP

Resumo. Redes neurais artificiais são utilizadas em diversos contextos, como classificação, controle inteligente, recuperação de informação, previsão do tempo, entre outros. O presente trabalho apresenta um modelo de agentes móveis reativos, que utiliza um controle baseado em redes neurais artificiais. O objetivo do modelo é fazer com que um grupo de agentes siga uma referência móvel e evite colisões com os agentes vizinhos. A criação do controle se deu via utilização de um algoritmo de otimização evolutivo, para redes neurais artificiais.

Palavras-chave. Agentes Móveis, Movimento Coletivo, Redes Neurais Artificiais, Algoritmo Evolutivo.

1 Introdução

Uma das aplicações de redes neurais artificiais é a criação de controles inteligentes, como ilustra o trabalho de Dierks *et al.* [2], em que os autores implementaram modelos para robôs móveis, em um esquema líder-seguidor, e com baixa troca de informação. Li e Liu [6] abordaram o problema de controle de robôs quadrúpedes, e Habibi *et al.* [5] trabalharam com o problema de transporte de um objeto por múltiplos robôs trabalhando coletivamente. Duarte *et al.* [3] desenvolveram controles para robôs aquáticos realizarem algumas tarefas, utilizando redes neurais geradas a partir do algoritmo evolutivo NEAT (*Neuroevolution of Augmenting Topologies*) [8].

O presente artigo apresenta um modelo de agentes móveis reativos, em que cada agente possui um controle baseado em redes neurais. O referido controle tem como entrada os dados normalizados de sensores simulados pelos agentes, e suas saídas são a velocidade angular e o momento linear. O objetivo dos agentes é seguir uma referência móvel, aqui chamado de *agente virtual* (AV), o qual aponta o caminho que o grupo de agentes deve

[1]vander.freitas@inpe.br

[2]quiles@unifesp.br

[3]elbert.macau@inpe.br

seguir [4]. O modelo, portanto, é composto por múltiplos agentes, os quais representam veículos artificiais se movendo em um plano 2D, e pelo agente virtual. Esse tipo de modelo pode ser aplicado em tarefas de exploração, monitoramento e coleta de dados, utilizando veículos aéreos não tripulados (VANTS), veículos aquáticos, robôs móveis.

A implementação foi realizada utilizando o algoritmo evolutivo NEAT [8] para geração automática e simulação da rede neural. O NEAT parte de uma topologia de rede simples e a evolui via operações evolutivas, a fim de otimizar uma dada função objetivo.

No trabalho de Duarte *et al.* [3], criou-se um controle para agrupar os agentes ao redor de uma determinada coordenada GPS. A ideia deste trabalho é semelhante, mas com a diferença de que essas coordenadas, aqui representadas pelo AV, mudam com o tempo.

2 NEAT

No contexto de otimização evolutiva, existem algoritmos consolidados, como os Algoritmos Genéticos (AG) [1] e o GEO (*Generalized Extremal Optimization*) [7]. Neles, prevalece a ideia de que ao longo das gerações a população deve melhorar, a fim de se adaptar ao meio, da melhor forma possível. Cada indivíduo da população contém as entradas para o cálculo da função objetivo, e eles sofrem modificações ao longo do tempo, no sentido de maximização (ou minimização) da função. Essas modificações ocorrem via operações como mutação e cruzamento, as quais alteram características de um indivíduo, ou combinam indivíduos para a geração de novos. O objetivo final, portanto, é encontrar os indivíduos mais bem adaptados à função objetivo escolhida.

O algoritmo NEAT (*Neuroevolution of Augmenting Topologies*) gera redes neurais artificiais automaticamente, e as evolui de forma a otimizar uma determinada função objetivo, sendo cada elemento da população uma rede neural. O algoritmo parte de uma rede com topologia mínima, e o processo de evolução se utiliza de mutação, cruzamento e especiação. Por topologia mínima, entende-se que a rede inicia com apenas duas camadas, a de entrada e a de saída. A quantidade de neurônios dessas duas camadas é igual ao número de parâmetros de entrada e valores de saída do controle abstraído pela rede.

Existem duas operações relacionadas à mutação, que são a adição de uma nova conexão ou de um novo nó na rede. No primeiro caso, uma aresta é inserida entre dois nós, caso entre eles não exista uma conexão prévia, dada uma probabilidade de mutação (PM). Para a mutação relacionada à adição de um nó, escolhe-se aleatoriamente uma aresta da rede, e o novo nó é inserido entre os dois nós das extremidades da referida aresta. A operação de cruzamento (*crossover*) dá origem a um novo indivíduo da população, partindo de dois indivíduos pais.

A especiação é um artifício empregado no NEAT para proteger inovação, isto é, conservar modificações topológicas recentes. Quando um novo nó - ou aresta - é inserido na rede, ela não está com os pesos adequadamente ajustados, e portanto pode ser que essa alteração degrade o resultado esperado. Para evitar que esse indivíduo seja instantaneamente excluído da população, ele é inserido em um grupo (espécie) em que todos os indivíduos possuem características parecidas. Nesse caso, a competição se torna mais justa, pois ele tem a chance de se desenvolver por algumas gerações.

DOI: 10.5540/03.2018.006.01.0345

3 Modelo de agentes móveis reativos

Os agentes são chamados de reativos, pois reagem a estímulos vindos de outros agentes, e não guardam informação de interações anteriores. Os componentes do modelo são os agentes reativos e um AV. Esse último representa um caminho a ser percorrido pelo grupo. Ele abstrai um sinal que se move a velocidade constante, podendo ser coordenadas GPS de terreno, por exemplo. Espera-se que os agentes iniciem em posições distantes, interajam entre si por meio do controle baseado em redes neurais, e sigam o AV.

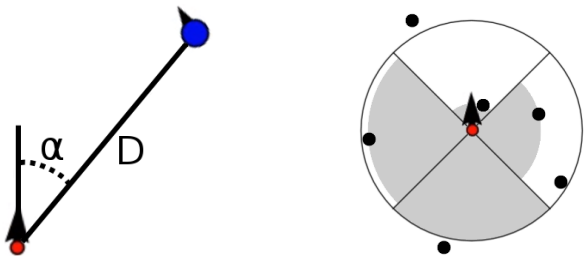
A dinâmica de um agente é dada pela Equação 1,

$$\begin{aligned} \dot{x}_i &= s_i \cos(\theta_i) \\ \dot{y}_i &= s_i \sin(\theta_i) \\ \dot{\theta}_i &= \text{net}_{\dot{\theta}_i} \\ \dot{s}_i &= \text{net}_{\dot{s}_i} \end{aligned} \tag{1}$$

onde $[x_i, y_i]^T \in \mathbb{R}^2$ é a posição do i-ésimo agente, θ_i é seu ângulo de navegação e s_i é o módulo da velocidade. Tanto $\dot{\theta}$ quando $\dot{s}$ são obtidos pelas saídas da rede neural.

Cada iteração (it) do modelo corresponde ao tempo em que o AV se move a uma distância equivalente ao seu tamanho (c), sendo a velocidade do AV igual a $s_{virtual} = 1c/it$.

Os agentes simulados interagem uns com os outros de acordo com os sensores da Figura 1. Na parte esquerda da imagem, D representa a distância entre o agente e o AV, e α é a diferença entre o ângulo de navegação do agente e a posição do AV. A segunda metade da Figura 1 representa o sensor de distâncias, o qual foi discretizado em quadrantes, e tem como saída a distância entre o vizinho mais próximo em cada quadrante.



(a) (b)

Figura 1: Sensores de percepção. (a) Sensor de navegação: determina a distância D entre o agente e o agente virtual, e a diferença entre o ângulo que ele está se movendo e a posição do agente virtual α. (b) Sensor de distâncias: define as distâncias ($Q1$, $Q2$, $Q3$ e $Q4$) entre o agente em questão e os seus vizinhos mais próximos em cada quadrante.
Fonte: Adaptado de [3]

O controle abstraído pela rede neural recebe os sinais normalizados dos sensores da Figura 1 como entrada (α, D, $Q1$, $Q2$, $Q3$ e $Q4$), e produz como saída a velocidade angular $\dot{\theta}$ e momento linear $\dot{s}$. Em [4], as interações entre agentes são realizadas de acordo com forças atrativas e repulsivas, descritas explicitamente. Por outro lado, a ideia aqui é que a rede neural aprenda as interações por meio de técnicas evolutivas.

DOI: 10.5540/03.2018.006.01.0345

Para geração da rede neural, foi utilizada uma função objetivo, descrita pela Equação 2, que leva em conta a distância entre os agentes e o AV, e também um termo S que penaliza a função na iminência de uma ou mais colisões, no decorrer da simulação [3]. O sentido de maximização da função depende de os agentes chegarem o mais próximo possível do AV, e evitarem colisões entre si.

$$f = \left(\frac{1}{T} \sum_{t=1}^{T} \frac{1}{R} \sum_{r=1}^{R} \frac{distInicial_r - dist_{r,t}}{distInicial_r} \right) \times S \ , \tag{2}$$

sendo R o número de agentes, T o tempo total da simulação, $distInicial_r$ a distância inicial entre o agente r e o AV, e $dist_{r_t}$ a distância entre o agente r e o AV no instante t.

O termo S da função objetivo é dado pela Equação 3

$$S = 0.1 + \frac{\max(0, \min(3, minDist))}{3} \times 0.9 \ , \tag{3}$$

em que $minDist$ é a mínima distância entre dois agentes, observada durante a simulação do modelo, dentro do tempo T. Definiu-se $3c$ como a distância mínima permitida, para evitar colisões, a qual corresponde à distância de três vezes o tamanho c de um agente. Caso durante toda a simulação os agentes obedeçam a distância mínima de $3c$, então S é igual a 1. Por outro lado, se em algum momento um par de agentes atinge uma distância menor do que $3c$, então $S < 1$, pois os agentes estiveram prestes a colidir.

4 Resultados

A geração do controle baseado em redes neurais considerou uma configuração do algoritmo NEAT com 50 indivíduos na população, 1500 gerações de evolução da população, e 5000 iterações do modelo reativo para cada configuração de rede.

Utilizou-se probabilidade de mutação $PM = 0,005$, tanto para adição de novos nós na rede, quanto conexões, e inicializou-se o NEAT com a topologia mais simples possível, de 9 neurônios, sendo 7 na camada de entrada e 2 na camada de saída. Os 7 neurônios na camada de entrada correspondem às seis entradas do controle (α, D, $Q1$, $Q2$, $Q3$ e $Q4$), e a uma entrada de bias, cujo valor é sempre 1. Os dois neurônios da camada de saída são relativos à velocidade angular $\dot{\theta}$ e momento linear $\dot{s}$.

Para avaliar os resultados, são utilizados dois índices, sendo o primeiro o índice de uniformidade radial, que corresponde à média das distâncias entre o agente virtual e cada agente da simulação. O segundo índice é o parâmetro de ordem de Kuramoto [9], definido pela equação 4.

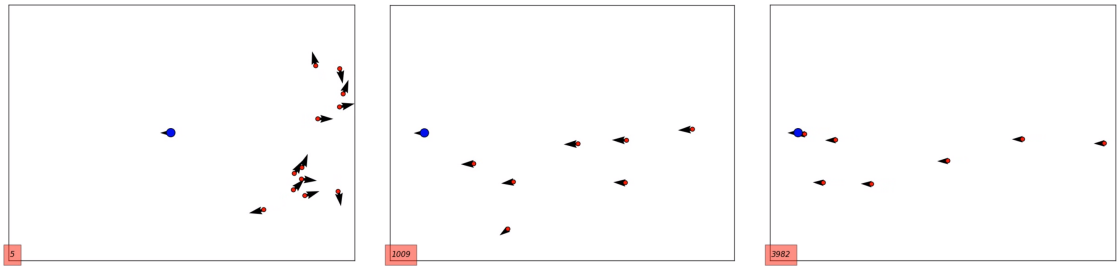
$$p_\theta \doteq \frac{1}{N} \sum_{j=1}^{N} e^{i\theta_j} \ , \tag{4}$$

sendo $e^{i\theta_k} = \cos\theta_j + i \sin\theta_j$, e θ_j o ângulo de navegação do j-ésimo agente. $|p_\theta|$ é igual a 1 quando os agentes possuem o mesmo ângulo de navegação, e $|p_\theta|$ tende a zero quando os ângulos se cancelam entre si, em direções opostas. Espera-se, portanto, que quanto

DOI: 10.5540/03.2018.006.01.0345

mais o módulo do parâmetro de ordem se aproxima de 1, mais paralelo é o movimento dos agentes.

Após executadas as 1500 gerações, a rede obtida produz resultados de simulação semelhantes aos da Figura 2. O fato de os agentes permanecerem com esta distância de segurança durante toda a simulação, indica que o controle é robusto a colisões, apesar de elas acontecerem em alguns raros casos. A rede neural correspondente é a apresentada na Figura 3.



(a) $t = 5it$ (b) $t = 1009it$ (c) $t = 3982it$

Figura 2: Simulação com o controle gerado pelo algoritmo NEAT.

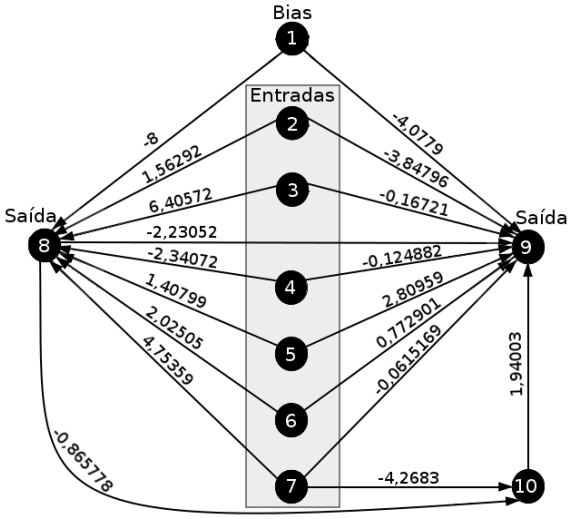


Figura 3: Rede gerada pelo algoritmo NEAT.

O controle obtido nas simulações foi submetido a testes com ruído aditivo aleatório ϵ, no cálculo do $\dot{\theta}$, como segue

$$\dot{\theta}_i = \text{net}_{\dot{\theta}_i} + \epsilon \ , \tag{5}$$

sendo $\epsilon \in [-\delta, \delta]$ e $\delta \in [0, 1]$. O modelo foi simulado 50 vezes com cada intensidade δ de ruído, variando δ de $\Delta\delta = 0,02$. Verificou-se que o modelo é robusto à presença de ruído no cálculo da dinâmica do ângulo de navegação dos agentes, resultando em

valores próximos no parâmetro de ordem, apesar da perturbação inserida. Houve apenas um pequeno decréscimo nos valores da função objetivo e um acréscimo na uniformidade radial, provavelmente pelo fato de os agentes terem mais dificuldade de se alinharem na direção do agente virtual. A Figura 4 apresenta uma média dos valores do parâmetro de ordem e uniformidade radial durante todos os instantes de simulação, e também o valor da função objetivo obtido ao fim da simulação. Os pontos em cinza representam os resultados obtidos nas 50 vezes em que o modelo foi simulado para cada intensidade δ do ruído, e os pontos em vermelho indicam o caso médio.

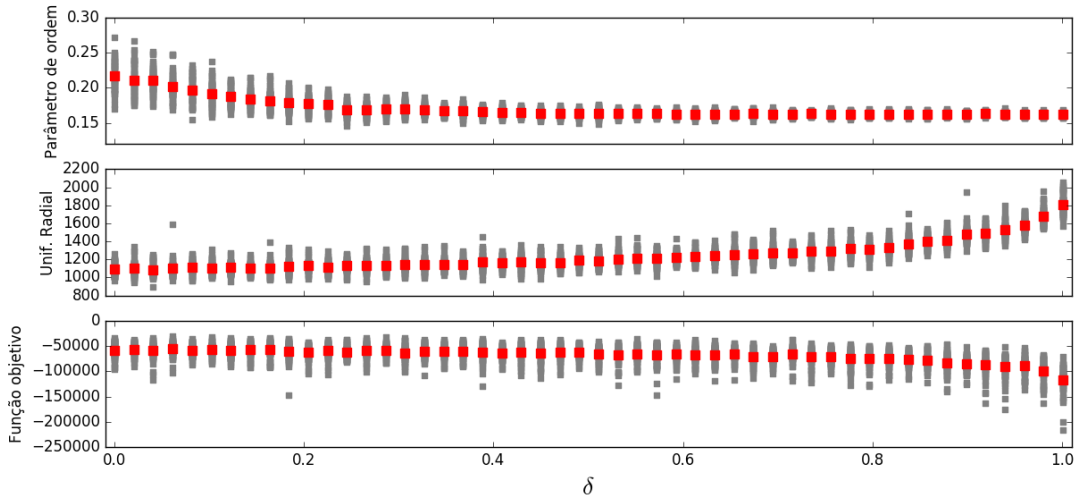


Figura 4: Simulações com ruído aleatório $\epsilon \in [-\delta, \delta]$. Resultados das simulações em cinza, e caso médio em vermelho.

Os valores de parâmetro de ordem obtidos no caso médio encontram-se entre $0,15$ e $0,20$, indicando que os agentes apresentam problemas para se manterem em formações paralelas durante todo o tempo. Isto ocorre devido à capacidade de evitar colisões, o que ocasiona mudanças de direção quando os agentes estão próximos uns dos outros. A função objetivo apresentou valores negativos devido às condições iniciais, dado que os agentes iniciam próximos do AV, mas com o passar do tempo eles se distanciam (Figura 2) por conta do controle de colisões. Caso a distância mínima permitida entre agentes fosse diminuída para valores menores que $3c$, a função objetivo apresentaria resultados mais positivos. Os padrões de formações obtidas se mostraram bastante estáveis, dado que o índice de uniformidade radial e o parâmetro de ordem se mantiveram quase constantes.

5 Conclusões

Este trabalho apresentou um modelo de agentes móveis reativos, no qual objetivou-se fazer com que eles seguissem uma referência móvel, chamada de agente virtual. A estratégia de controle utilizada baseou-se em redes neurais artificiais, tendo como entrada os dados normalizados dos sensores simulados pelos agentes, e como saída a velocidade

DOI: 10.5540/03.2018.006.01.0345

angular e o momento linear.

A rede neural que representa o controle foi obtida por meio de um algoritmo evolutivo para geração automática de redes neurais, o NEAT. Com esse algoritmo, é possível gerar controles que satisfaçam uma dada função objetivo. O sentido de maximização da função escolhida, preza pelo agrupamento dos agentes reativos nas proximidades do agente virtual, e conta com um mecanismo para evitar colisões.

O controle obtido mostrou-se robusto à presença de ruído aditivo no ângulo de navegação dos agentes, tendo resultados similares ao caso sem ruído. Como trabalho futuro, espera-se experimentar o modelo com robôs móveis.

Agradecimentos

Os autores gostariam de agradecer ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) e a Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) pelo apoio financeiro. EENM agradece à Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP), processo 2015/50122-0, e CNPq, processo 458070/2014-9.

Referências

[1] T. Bäck and H.-P. Schwefel, An Overview of Evolutionary Algorithms for Parameter Optimization, Evol. Comput., vol. 1, no. 1, pp. 1-23, 1993.

[2] T. Dierks, B. Brenner, and S. Jagannathan, Neural Network-Based Optimal Control of Mobile Robot Formations With Reduced Information Exchange, IEEE Trans. Control Syst. Technol., vol. 21, no. 4, pp. 1407-1415, 2013.

[3] M. Duarte et al., Evolution of Collective Behaviors for a Real Swarm of Aquatic Surface Robots, PLoS One, vol. 11, no. 3, pp. 1-25, 2016.

[4] V. L. S. Freitas and E. E. N. Macau, Reactive Agent-Based Model for Convergence of Autonomous Vehicles to Parallel Formations Heading to Predefined Directions of Motion, in Proceedings of the 9th International Conference on Agents and Artificial Intelligence, 2017, vol. 1.

[5] G. Habibi, Z. Kingston, W. Xie, M. Jellins, and J. McLurkin, Distributed centroid estimation and motion controllers for collective transport by multi-robot systems, in 2015 IEEE International Conference on Robotics and Automation (ICRA), 2015, pp. 1282-1288.

[6] Z. Li, S. S. Ge, and S. Liu, Contact-Force Distribution Optimization and Control for Quadruped Robots Using Both Gradient and Adaptive Neural Networks, IEEE Trans. Neural Networks Learn. Syst., vol. 25, no. 8, pp. 1460-1473, 2014.

[7] F. L. Sousa, F. M. Ramos, P. Paglione, and R. M. Girardi, New Stochastic Algorithm for Design Optimization, AIAA J., vol. 41, no. 9, pp. 1808-1818, Sep. 2003.

[8] K. O. Stanley and R. Miikkulainen, Evolving Neural Networks through Augmenting Topologies, Evol. Comput., vol. 10, no. 2, pp. 99-127, Jun. 2002.

[9] S. H. Strogatz, From Kuramoto to Crawford: Exploring the Onset of Synchronization in Populations of Coupled Oscillators, Phys. D, vol. 143, no. 1-4, pp. 1-20, 2000.

