Application of Improved PSO Algorithm in Multi-UAV Cooperative Task Allocation Postprint

Authors: Jiang Shuo, Yuan Xiaoping

Date: 2018-08-13T00:00:00+00:00

Abstract

To address the increasingly complex problem of multi-UAV cooperative task allocation, an improved hierarchical particle swarm optimization algorithm (HGI-WPSO) is adopted to obtain the optimal allocation solution. First, the population is dynamically partitioned into three distinct hierarchies based on particle fitness values; appropriate learning models are selected according to the characteristics of particles in different hierarchies, and the concept of independent weight is introduced to regulate the inertia weight magnitude, balancing the algorithm' s global and local search capabilities and enhancing its performance. Subsequently, a cooperative multi-task allocation problem model is constructed; a surplus load auction scheme is employed to reduce illegal inferior solutions, and a mapping relationship between particles and actual allocation schemes is established through real-number encoding to address the practical allocation problem. Experimental results indicate that the proposed algorithm can effectively solve the multi-UAV cooperative task allocation problem under complex constraints, yielding the optimal allocation sequence, and holds certain theoretical and practical significance.

Full Text

Preamble

Application of Improved PSO Algorithm in Multi-UAV Cooperative Task Allocation

Jiang Shuo, Yuan Xiaoping

(China University of Mining & Technology, Xuzhou, Jiangsu 221116, China)

Abstract: To address the increasingly complex problem of multi-UAV cooperative task allocation, this paper employs an improved hierarchical particle swarm optimization algorithm (HGIWPSO) to obtain optimal allocation schemes. First, the population is dynamically divided into three distinct classes based on particle fitness values. Appropriate learning models are selected according to the characteristics of particles in different classes, and the concept of independent weight is introduced to adjust the inertia weight, balancing global and local search capabilities and improving algorithm performance. Then, a collaborative multi-task allocation problem model is established, and a redundant load auction scheme is adopted to reduce illegal inferior solutions. Real-number encoding is used to establish the mapping relationship between particles and actual allocation schemes to solve practical allocation problems. Experimental results demonstrate that the algorithm can effectively solve multi-UAV cooperative task allocation problems under complex constraints, obtain optimal allocation sequences, and possesses both theoretical and practical significance.

Keywords: UAV; hierarchical classification; task assignment; particle swarm optimization; real-number encoding

0 Introduction

Current research on UAV task allocation problems both domestically and internationally primarily focuses on two aspects: mathematical modeling and optimization of allocation models. The main achievements in modeling include the Vehicle Routing Problem (VRP) model, Multiple Traveling Salesman Problem (MTSP) model, Mixed-Integer Linear Programming (MILP) model, Network Flow Optimization (NFO) model, and Cooperative Multi-Task Allocation Problem (CMTAP) model. Literature [1] treats UAVs as vehicles in the VRP model and reconnaissance targets as passengers, thereby simplifying the UAV path planning problem. Literature [2] decomposes the UAV task allocation problem into classical MTSP and NFO problems for solution. Literature [3, 4] establishes MILP models for multiple UAVs by constructing constraint conditions and objective functions. Literature [5] analyzes constraint relationships between various task types, optimizes total UAV flight distance and required time, and establishes a CMTAP model.

The main algorithms for solving these models include Tabu Search, Genetic Algorithm (GA), Ant Colony Optimization (ACO), and Particle Swarm Optimization (PSO). Literature [12] sets up tabu lists to characterize existing local optimal positions, continuously updating and testing local optimal values during the solution process to ultimately obtain the optimal solution. Literature [13] designs a genetic algorithm to solve multi-UAV cooperative allocation problems, and comparison with branch-and-bound and random search methods demonstrates its superior time efficiency. Literature [14] achieves discrete encoding of UAVs through binary matrix encoding, then enhances particle diversity through crossover and mutation strategies, effectively applying the PSO algorithm to solve multi-UAV reconnaissance task allocation problems. Literature

[15] proposes an integrated control architecture for multi-UAV task allocation and path planning, using a particle-improved PSO algorithm matched with the cooperative system to solve task allocation problems.

While these methods can solve UAV cooperative task allocation problems to a certain extent, they suffer from issues such as algorithm complexity or low solution efficiency. The PSO algorithm, with fewer parameters, simple principles, and fast search speed, is suitable for solving UAV cooperative task allocation problems. This paper proposes a hierarchical classification strategy that provides corresponding update methods for particles of different classes based on their characteristics, enhances population diversity, improves algorithm convergence accuracy and search efficiency, and employs real-number encoding to simplify the algorithm for solving UAV cooperative task allocation problems, achieving favorable results.

1.1 Model Description

The primary environment considered in this paper is a simulated two-dimensional battlefield scenario. The target task set in the battlefield is denoted as $\{T_1, T_2, ..., T_{N_t}\}$, where N_t represents the total number of tasks to be executed. We use a six-tuple $(T_{id}, T_{Pos}, T_{State}, T_{Value}, T_D, T_{Thr})$ to characterize the attributes of each task target, where each task has a unique identifier T_{id}. In the two-dimensional battlefield space, T_{Pos} represents its coordinate position, T_{State} represents the current status of the task target, T_{Value} represents the value of the UAV attack on the specified task target (i.e., the target value), T_D represents the threat range area of the enemy target, and T_{Thr} represents the threat coefficient to the UAV itself when attacking the enemy target (i.e., the probability of the UAV being shot down while executing the task), with $T_{Thr} \in [0, 1)$.

The UAV set in this paper is $\{U_1, U_2, ..., U_{N_u}\}$, where N_u represents the number of UAVs. UAV attributes are represented by a seven-tuple $(U_{id}, U_{Pos}, U_{State}, U_{Value}, U_{At}, Ts, L)$, with contents similar to task targets, including the UAV number executing the target, the UAV' s coordinate position in the two-dimensional battlefield, current status, the UAV' s own value, the UAV' s attack capability on the target, the task set, and the maximum task execution load. $U_{At} \in [0, 1)$ represents the probability that the UAV can destroy the task target when attacking it, and L represents the maximum number of tasks the UAV can execute, with $1 \leq L_i \leq L_S \leq N_t$.

Multi-UAV cooperative allocation is essentially a multi-objective optimization problem. This paper transforms the complex multi-objective optimization problem into a relatively simple single-objective optimization problem by referencing and drawing upon the linear weighted sum method. A weight vector $\omega = (\omega_1, \omega_2, \omega_3)$ is set to balance the influence degree of various factors on the allocation result, satisfying $\omega_i \in (0, 1)$ and $\sum_{i=1}^{3} \omega_i = 1$. The multi-UAV task allocation problem is converted into solving the optimization problem shown in Equation (6):

$$\min f = \sum_{i=1}^{N_u} \sum_{j=1}^{N_t} (\omega_1 C_1 + \omega_2 C_2 - \omega_3 C_3)\, x_{ij}$$

The constraint conditions are shown in Equations (7) to (9):

$$\sum_{i=1}^{N_u} x_{ij} = 1, \quad \forall j \in J$$

$$\sum_{j=1}^{N_t} x_{ij} \leq L_i, \quad \forall i \in I$$

$$\min\{N_u L_i, N_t\} \geq \sum_{i=1}^{N_u} \sum_{j=1}^{N_t} x_{ij} \geq N_t$$

where the decision variable x_{ij} is a 0-1 variable, with $x_{ij} = 1$ indicating that UAV U_i executes task T_j, meaning UAV U_i can destroy task target T_j when attacking it. $I = \{1, 2, ..., N_u\}$ and $J = \{1, 2, ..., N_t\}$ represent the index sets for UAVs and targets, respectively.

1.2 Cost Function

During the process of UAVs attacking enemy targets, the probability of being destroyed by hostile targets should be minimized as much as possible. Let the survival probability of the i-th UAV after completing task T_j be p_{ij}. Therefore, the cost value required for a UAV to complete N_i tasks is shown in Equation (1):

$$C_1 = U_i^{Value} \left(1 - \prod_{j=1}^{N_i} (1 - Thr_j) \right)$$

where U_i^{Value} represents the value of the UAV, and Thr_j represents the probability of the UAV being shot down by hostile target T_j.

The distance between the UAV executing the attack task and the target task point determines the magnitude of the flight distance cost. The closer the UAV is to the target, the smaller the flight distance cost will be, and thus the target has a higher probability of being executed by a nearby UAV. The expression for the flight distance cost is shown in Equation (2):

$$C_2 = \frac{d_{ij}}{d_{max}}$$

where d_{ij} represents the Euclidean distance between UAV U_i and target T_j, and d_{max} represents the maximum distance among all UAVs from the target, i.e., $d_{max} = \max(d_{i,j})$.

Attack benefit refers to the target value obtained by the UAV after completing the attack task and destroying the target. This evaluation indicator guides the final result toward maximizing combat effectiveness in the optimization decision-making for targets, as shown in Equation (3):

$$C_3 = U_i^{At} \times T_j^{Value}$$

Since the three evaluation criteria mentioned above are not in unified units of measurement and have non-commensurability, they cannot be simply added and integrated. They require standardization transformation through certain methods. This paper selects the linear scale transformation method to normalize the UAV value and target value comparison benefit indicators, as shown in Equations (4) and (5):

$$C_1' = \frac{C_1}{\max_{i \in N_u}(U_i^{Value})}$$

$$C_3' = \frac{C_3}{\max_{j \in N_t}(T_j^{Value})}$$

2.1 Basic PSO Algorithm

The mathematical description of the basic PSO algorithm is as follows. The initial population size of the particle swarm is set to M, and the particle dimension is D dimensions. At time t, the coordinates of particle i in the solution space are defined as $X_i^t = (x_{i1}^t, x_{i2}^t, ..., x_{iD}^t)$, where $i = 1, 2, ..., M$. Its velocity, which determines the particle' s flight distance in the solution space each time, is denoted as $V_i^t = (v_{i1}^t, v_{i2}^t, ..., v_{iD}^t)$. The velocity and position update formulas for particle i at time t in the j-th dimension are shown in Equations (10) to (12):

$$v_{ij}^{t+1} = \omega v_{ij}^t + c_1 r_1 (p_{ij} - x_{ij}^t) + c_2 r_2 (g_j - x_{ij}^t)$$

$$v_{ij}^{t+1} = \begin{cases} v_{max}, & v_{ij}^{t+1} > v_{max} \\ -v_{max}, & v_{ij}^{t+1} < -v_{max} \end{cases}$$

$$x_{ij}^{t+1} = x_{ij}^{t} + v_{ij}^{t+1}$$

where ω is the inertia weight, c_1 and c_2 are acceleration constants, and r_1 and r_2 are two random numbers in $[0, 1]$. During the search process in the solution space, particle velocity is limited to a maximum search range, with the maximum velocity denoted as v_{max} to constrain the particle' s search range and improve algorithm performance. g_j represents the historical optimal position of the group during iteration, i.e., the global extremum g_{best}, while p_{ij} represents the optimal position of the particle itself during its search process, i.e., the individual extremum p_{best}.

2.2 Hierarchical PSO Algorithm

In the PSO optimization process, the fitness values of all particles in the current generation are represented by the set $F = \{f_1, f_2, ..., f_m\}$, where m represents the total number of particles and f_i represents the fitness value of the i-th particle. Based on the calculated fitness values, all particles are sorted in ascending order (where smaller fitness values represent better particle positions) to form a new particle sequence set $X = \{x_1, x_2, ..., x_m\}$. The population is divided into three different classes by judging the distance of particles from the first particle. Let the sequence number of the globally optimal particle after reordering be g_N. The specific hierarchical classification strategy is described in detail as follows:

If $i \leq N/4$, then the particle is classified as an upper-class particle. For upper-class particles, their own search is already near the optimal value. As the algorithm iterates continuously, the particle itself gets closer to the optimal position. At this time, the experience of the entire group no longer has significant learning value for the particle itself. Therefore, the particle ignores group learning experience during iterative updates and mainly considers its own learning experience model. For this reason, we propose a new velocity update formula for upper-class particles based on the original velocity update formula to adapt to the particle' s own characteristics and achieve better search optimization. The velocity update formula for upper-class particles is shown in Equation (13):

$$v_{ij}^{t+1} = \omega v_{ij}^{t} + c_1 r_1 (p_{ij} - x_{ij}^{t})$$

If $N/4 \leq i \leq 3N/4$, then the particle is classified as a middle-class particle. During the search process, particles need to learn from both their own experience and group experience. Therefore, the velocity update method for middle-class particles maintains the standard PSO algorithm' s velocity update formula unchanged, as shown in Equation (14):

$$v_{ij}^{t+1} = \omega v_{ij}^t + c_1 r_1 (p_{ij} - x_{ij}^t) + c_2 r_2 (g_j - x_{ij}^t)$$

If $3N/4 < i \leq M$, then the particle is classified as a lower-class particle. For lower-class particles, their search position has already deviated far from the optimal position, and the value of their own search experience is limited. Therefore, lower-class particles adopt a group learning model for velocity updates, as shown in Equation (15):

$$v_{ij}^{t+1} = \omega v_{ij}^t + c_2 r_2 (g_j - x_{ij}^t)$$

Since lower-class particles have already deviated far from the optimal solution position, to enable particles at this level to exchange experience information with excellent particles, lower-class particles are crossed with the global optimal particle at a certain probability p. Two points a and b are randomly selected, satisfying $a, b \in [1, D]$ and $a < b$, where D represents particle dimension. The dimensions between them are crossed to form new particles.

The PSO algorithm tends to lose diversity in the later stages of iteration, causing the algorithm to fall into local optima. To solve this problem, random disturbance velocity is introduced in the algorithm to escape local traps. The dimensional distance of the i-th particle to the current best particle is defined as d_i, and the standard deviation of dimensional distance is shown in Equation (17):

$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (d_i - \bar{d})^2}$$

where $\bar{d}$ represents the expected value of dimensional distance. When the particle state satisfies $\sigma < \varepsilon$ (where ε is a defined nominal value), it is determined that the algorithm has fallen into a local optimal trap during iteration. At this point, a random velocity disturbance is given to help particles escape the local optimum and continue iterative search. The random disturbance velocity update formula is shown in Equation (18):

$$v_{ij}^{t+1} = \omega v_{ij}^t + c_1 r_1 (p_{ij} - x_{ij}^t) + c_2 r_2 (g_j - x_{ij}^t) + \mathrm{rand}()$$

Combining the characteristics of the improvement strategy in this paper, independent inertia weight is selected to adjust the particle' s global and local optimization capabilities, as shown in Equation (19):

$$\omega_i(t) = \omega_{start} - (\omega_{start} - \omega_{end}) \cdot e^{-\kappa_i \cdot t/N}$$

where ω_{start} is the initial weight value, ω_{end} is the final weight value, and κ_i is the exponential change factor, calculated as follows:

$$\kappa_i = \arctan(\log(\phi \cdot |\mathrm{fit}_i - \overline{\mathrm{fit}}|)) \cdot \eta \cdot \pi$$

where ϕ and η are fixed constants, and fit_i and $\overline{\mathrm{fit}}$ represent the fitness value of particle i and the average fitness value, respectively. Mathematical transformation of the data is performed to satisfy assumptions as much as possible and ensure relatively stable fluctuations. t represents the current iteration number of the particle.

---

3 HGIWPSO Algorithm for Task Allocation

3.1 Algorithm Performance

Standard test functions are used to evaluate algorithm performance, as shown in . The algorithm' s optimization performance after improvement is evaluated by statistically analyzing the mean and standard deviation of fitness error values and studying the variation curve of fitness values with iteration times. Taking the average of 500 runs, the simulation results are as follows.

shows the test functions and their related attributes used in the experiments, including Sphere, Rosenbrock, Ackley, Griewank, and Rastrigin functions. and present the algorithm comparison data for 10-dimensional and 30-dimensional cases, respectively. The data clearly demonstrate that whether for high-dimensional or low-dimensional situations, the HGIWPSO algorithm can jump out of local optimal positions compared to other improved algorithms, achieving better optimization accuracy and convergence speed. The error and mean square error are superior to other improved algorithms, proving the feasibility and effectiveness of the improvement.

The convergence comparison curves for various functions are shown in [Figure 1: see original paper] through [Figure 10: see original paper], illustrating the performance across different dimensions and test functions.

3.2 HGIWPSO Algorithm Implementation for Task Allocation

This paper adopts real-number encoding for the task allocation problem. The number of attacked task targets equals the particle dimension. The integer part of the particle is used to identify the UAV number executing the task, with the same integer part representing the encoding sequence of tasks executed by one UAV. The decimal part is arranged in ascending order to determine the sequence of attack tasks executed by the UAV. An auction mechanism is introduced to optimize and reduce illegal inferior solutions, ensuring load constraint conditions can be satisfied.

The execution flow of the task allocation algorithm with auction mechanism is as follows:

a) Initialize the particle swarm;

b) Decode the initialized particles. If constraints are satisfied, proceed to the next step; if not, adjust and correct according to the task coordination allocation method of the auction algorithm, re-encode particles, and update particle positions;

c) Calculate the fitness values of each particle;

d) Update particle velocities and positions;

e) Check whether convergence criteria are met. If met, output the allocation scheme and terminate the algorithm; if not, return to step b) and continue execution.

To verify the effectiveness of the algorithm in task allocation applications, simulation comparison experiments are conducted. The parameters for the algorithm simulation are as follows: population size is 40, particle dimension is 7, maximum iteration number is 300, tactical weights are (0.2, 0.4, 0.4), fixed constants are 0.73 and 1.02, and maximum and minimum inertia weights are 0.9 and 0.4, respectively. The average of 50 runs is taken.

Target parameters for three UAVs executing tasks and the parameters of attack targets are shown in and . shows the task allocation results. The simulation results demonstrate that as algorithm iteration progresses, the total cost of executing tasks continuously decreases. The comparison experiments show that PSO, SAPSO, and CLSPSO algorithms fall into local optima around 200-250 generations, while the improved algorithm proposed in this paper can jump out of local positions in UAV task allocation and ultimately find the optimal position (2.7473, 1.8132, 1.4449, 3.6773, 2.7441, 1.7111, 3.8870), minimizing the total task execution cost to 0.8433. The algorithm converges to the optimal position around 170 generations with fast convergence speed, proving that the improved algorithm applied to multi-UAV cooperative task allocation achieves good results, and the improvement strategy is feasible with certain theoretical and practical significance.

---

4 Conclusion

This paper addresses the complex multi-coupling characteristics of multi-UAV cooperative task allocation, proposes a feasible modeling scheme, and simplifies the problem for solution. The improved PSO algorithm using hierarchical classification strategy combined with independent weight effectively solves the task allocation problem. The improved algorithm employs hierarchical classification and independent weight concepts, which differ from existing improvement strategies that mostly target the entire group. Instead, it fully reflects the characteristics of particles in different classes through hierarchical classification and balances individual particle global and local search capabilities through independent weight strategy. This effectively enhances particle diversity, enables particles to more easily escape local optima, accelerates search speed, well compensates for algorithm defects, and shows more significant effects compared to current related improved algorithms. Applying the improved algorithm to solve multi-UAV cooperative task allocation problems, using real-number encoding to reduce problem-solving difficulty and load auction method to solve practical load constraint problems, achieves good results.

References

[1] Zorlu O. Routing unmanned aerial vehicles as adapting to capacitated vehicle routing problem with genetic algorithms [C]// Proc of International Conference on Recent Advances in Space Technologies. 2015: 675-679.

[2] Kim M H, Baik H, Lee S. Response threshold model based uav search planning and task allocation [J]. Journal of Intelligent & Robotic Systems, 2014, 75 (3-4): 625-640.

[3] 叶媛媛, 闵春平, 朱华勇, 等. 基于整数规划的多 UCAV 任务分配问题研究 [J]. 信息与控制, 2005, 34 (5): 548-552. (Ye Yuanyuan, Min Chunping, Zhu Huayong, et al. Research on multi UCAV task allocation based on integer programming [J]. Information and Control, 2005, 34 (5): 548-552.)

[4] 张庆杰. 基于一致性理论的多 UAV 分布式协同控制与状态估计方法 [D]. 长沙: 国防科学技术大学, 2011. (Zhang Qinjie. Multi UAV distributed cooperative control and state estimation method based on consistency theory [D]. Changsha: National University of Defense Science and Technology, 2011.)

[5] Schumacher C, Chandler P R, Rasmussen S J, et al. Task allocation for wide area search munitions with variable path length [C]// Proc of the American Control Conference. 2003: 3472-3477.

[6] 于莹莹, 陈燕, 李桃迎. 改进的遗传算法求解旅行商问题 [J]. 控制与决策, 2014, 29 (8): 1483-1488. (Yu Yingying, Chen Yan, Li Taoying. Improved genetic algorithm for traveling salesman problem [J]. Control and Decision, 2014, 29 (8): 1483-1488.)

[7] 刘全, 王晓燕, 傅启明, 等. 双精英协同进化遗传算法 [J]. 软件学报, 2012, 23 (4): 765-775. (Liu Quan, Wang Xiaoyan, Fu Qiming, et al. Double elitist coevolutionary genetic algorithm [J]. Journal of Software, 2012, 23 (4): 765-775.)

[8] 金敏, 鲁华祥. 一种遗传算法与粒子群优化的多子群分层混合算法 [J]. 控制理论与应用, 2013, 30 (10): 001231-1238. (Jin Ming, Nu Huaxiang. A multi subgroup hierarchical

hybrid algorithm based on genetic algorithm and particle swarm optimization [J]. Control Theory and Application, 2013, 30 (10): 001231-1238.)

[9] 王磊, 郭顺生, 李西兴, 等. 基于改进蚁群算法的多主体制造资源配置冲突协调方法 [J]. 计算机集成制造系统, 2017 (12). (Wang Lei, Guo Shunsheng, Li Xixing, et al. Conflict coordination method of multi-agent manufacturing resource allocation based on improved ant colony algorithm [J]. Computer Integrated Manufacturing System, 2017 (12).)

[10] 冯金芝, 陈兴, 郑松林. 一种改进的多目标粒子群优化算法及其应用 [J]. 计算机应用研究, 2014, 31 (3): 675-678. (Feng Jinzhi, Chen Xing, Zheng Songlin. An improved multi-objective particle swarm optimization algorithm and its application [J]. Application Research of Computers, 2014, 31 (3): 675-678.)

[11] 王曙燕, 杨悦, 孙家泽. 基于改进粒子群算法的变异体选择优化 [J]. 计算机应用研究, 2017, 34 (3): 752-755. (Wang Shuyan, Yang Yue, Sun Jiangze. Optimization of variant selection based on improved particle swarm optimization algorithm [J]. Application Research of Computers, 2017, 34 (3): 752-755.)

[12] Hua-Wei M A, Fan F W, Xiao-Xuan H U, et al. Solving the Multi-Truck and Single Trailer Routing Problem Based on Tabu Search Algorithm [J]. Chinese Journal of Management Science, 2016.

[13] Shima T, Rasmussen S J, Sparks A G, et al. Multiple task assignments for cooperating uninhabited aerial vehicles using genetic algorithms [J]. Computers & Operations Research, 2006, 33 (11): 3252-3269.

[14] Zhang Y Z, Li J W, Hu B, et al. An improved PSO algorithm for solving multi-UAV cooperative reconnaissance task decision-making problem [C]// Proc of IEEE International Conference on Aircraft Utility Systems. 2016.

[15] 孙小雷, 齐乃明, 董程, 等. 无人机任务分配与航迹规划协同控制方法 [J]. 系统工程与电子技术, 2015, 37 (12): 2772-2776. (Sun Xiaolei, Qi Naiming, Dong Chen, et al. Cooperative control method for UAV mission allocation and route planning [J]. System Engineering and Electronic Technology, 2015, 37 (12): 2772-2776.)

[16] 杜继永, 张凤鸣, 杨骥, 等. 多 UCAV 协同任务分配模型及粒子群算法求解 [J]. 控制与决策, 2012, 27 (11): 1751-1755. (Du Jiyong, Zhang Fengming, Yang Ji, et al. Multi UCAV cooperative task allocation model and particle swarm optimization algorithm [J]. Control and Decision, 2012, 27 (11): 1751-1755.)

[17] Masdari M, Salehi F, Jalali M, et al. A survey of PSO-based scheduling algorithms in cloud computing [J]. Journal of Network & Systems Management, 2017, 25 (1): 122-158.

[18] 刘振, 周先存. 基于独立权重和分级变异策略的粒子群算法 [J]. 吉林大学学报: 理学版, 2017, 55 (2): 333-339. (Liu Zheng, Zhou Xiancun. Particle swarm optimization algorithm based on independent weight and hierarchical mutation strategy [J]. Journal of Jilin University, 2017, 55 (2): 333-339.)

[19] Doğan B, Ölmez T. A new metaheuristic for numerical function optimization: Vortex Search algorithm [J]. Information Sciences, 2015, 293: 125-145.

Note: Figure translations are in progress. See original paper for figures.