DOI 10.15302/J-SSCAE-2018.04.017

群智进化理论及其在智能机器人中的应用

戚骁亚[1]，刘创[1]，富宸[1]，甘中学[2]

（1. 北京深度奇点科技有限公司，北京 100086；2. 复旦大学智能机器人研究院，上海 200433）

摘要：群体智能（CI）已经在过去的几十年里被广泛研究。最知名的 CI 算法就是蚁群算法（ACO），它被用来通过 CI 涌现解决复杂的路径搜索问题。最近，DeepMind 发布的 AlphaZero 程序，通过从零开始的自我对弈强化学习，在围棋、国际象棋、将棋上都取得了超越人类的成绩。通过在五子棋上试验并实现 AlphaZero 系列程序，以及对蒙特卡洛树搜索（MCTS）和 ACO 两种算法的分析和比较，AlphaZero 的成功原因被揭示，它不仅是因为深度神经网络和强化学习，而且是因为 MCTS 算法，该算法实质上是一种 CI 涌现算法。在上述研究基础上，本文提出了一个 CI 进化理论，并将其作为走向人工通用智能（AGI）的通用框架。该算法融合了深度学习、强化学习和 CI 算法的优势，使得单个智能体能够通过 CI 涌现进行高效且低成本的进化。此 CI 进化理论在智能机器人中有天然的应用。一个云端平台被开发出来帮助智能机器人进化其智能模型。作为这个概念的验证，一个焊接机器人的焊接参数优化智能模型已经在云端平台上实现。

关键词：群体智能；涌现；进化；正反馈；蚁群算法；蒙特卡洛树搜索；分布式人工智能云端平台；智能机器人

中图分类号：TP242.6　　**文献标识码：**A

Theory of Collective Intelligence Evolution and Its Applications in Intelligent Robots

Qi Xiaoya[1], Liu Chuang[1], Fu Chen[1], Gan Zhongxue[2]

(1. Beijing Deep Singularity Technology Co., Ltd., Beijing 100086, China; 2. Intelligent Robot Research Institute, Fudan University, Shanghai 200433, China)

Abstract: Collective intelligence (CI) is widely studied in the past few decades. The most well-known CI algorithm is the ant colony optimization (ACO). ACO is used to solve complex path searching problems through CI emergence. Recently, DeepMind announced the AlphaZero program which has achieved superhuman performance in the game of Go, Chess, and Shogi, by tabula rasa reinforcement learning from games of self-play. By experimenting and implementing the AlphaZero series program in the game of Gomoku, along with analyzing and comparing the Monte-Carlo tree search (MCTS) and ACO algorithms, it is realized that the success of AlphaZero is not only due to the deep neural network and reinforcement learning, but also due to the MCTS algorithm, which is discovered to be a CI emergence algorithm. Thus we propose a CI evolution theory, as a general framework towards artificial general intelligence (AGI). Combining the strengths of deep learning, reinforcement learning, and CI algorithm, CI evolution theory enables individual intelligence to evolve with high efficiency and low cost through CI emergence. This CI evolution theory has natural applications in intelligent robots. A cloud-terminal platform is developed to help intelligent robots evolve their intelligent models. As a proof of this idea, a welding robot's welding parameter optimization intelligent model is implemented on the platform.

Keywords: collective intelligence; emergence; evolution; positive feedback; ant colony optimization; Monte-Carlo tree search; distributed AI cloud-terminal platform; intelligent robot

收稿日期：2018-08-10；**修回日期：**2018-08-15

通讯作者：甘中学，复旦大学智能机器人研究院，教授，研究方向为智能制造领域、柔性自动化控制及能源系统控制工程技术；E-mail: zhongxuegan@126.com

资助项目：中国工程院咨询项目“新一代人工智能引领下的智能制造研究”(2017-ZD-08-03)

本刊网址：www.enginsci.cn

一、前言

群体智能（CI）的概念源自 1785 年 Condorcet 的陪审团定理：如果投票组的每个成员有超过一半的机会做出正确的决定，则组中多数决定的准确性随着组成员数目的增加而增加 [1]。在 20 世纪下叶，CI 被应用到机器学习领域 [2]，并对如何设计智能体的集合以满足全系统的目标进行了更广泛的考虑 [3,4]。这与单智能体的奖励成形有关 [5]，并在博弈论界和工程界得到了众多研究者的关注 [6]。然而，CI 算法，如众所周知的蚁群算法（ACO），关注如何使群体智能涌现并超越个体智能，缺乏进化个体智能的机制，因此在没有重大扩展的情况下不能成为自我进化的人工通用智能（AGI）体。

AGI 的一个长期目标是创建能够从第一原理进行自我学习的程序 [7]。最近，AlphaZero 算法通过使用深度卷积神经网络和自我对弈游戏中的强化学习，在围棋、国际象棋和将棋游戏中达到超人的性能。然而，AlphaZero 如此成功的原因还没有被完全理解。通过分析和试验 AlphaZero 可以感觉到群智智能的逻辑思维暗含在算法当中。

本文从 CI 的发展和逻辑思路出发，将 AlphaZero 算法应用到五子棋的博弈中，展现了深度神经网络的进化能力；然后，又将蒙特卡洛树搜索（MCTS）与 ACO 进行比较，识别出 MCTS 是一种 CI 算法。最后，在深入分析和系统综合的基础上，笔者提出了 CI 进化理论，将其作为走向 AGI 的通用框架，并将其应用于智能机器人的应用。

二、群体智能概述

近年来，CI 被广泛应用于各种工作中，如项目中的人员协作、公司董事会的投资决策、总统选举投票等。看起来一个群体做事比个体更聪明。然而，Bon 在他的著名著作《乌合之众》中指出，群体行为可能是极端的 [8]。在这个意义上，CI 不能通过个人的简单组合来实现，而应该首先理解 CI 的特征，以更好地利用它来实现我们的目标。

在社会学领域，麻省理工学院群智中心的一组研究者将需要的工作分为四个组成部分，即执行者、动机、目标和实现方式，并在此基础上提出“群智基因组”[9]。以谷歌和维基百科为例，分别对这些组织的基因进行分析，并提出“CI 基因”有用的条件。此外，他们的同事在两个不同的实验中系统地研究了团队的表现，并得出了衡量一个团队的一般能力的“C 因子”[10]。这个“C 因子”与群体成员的平均社会敏感度、话语权力的平等性以及女性在群体中的比例相关。可以预见的是，通过重组“CI 基因”，并根据任务的“C 因子”，人们可以得到一个他需要的强大系统。

在这些 CI 社会学理论的基础上，人们可以在群体力量的帮助下更好地解决问题，尤其是在计算机科学中。1991 年，Colorni 等人 [11] 研究蚂蚁的食物搜索行为，并提出蚁群算法 [12~14]。该算法的基本思想是基于信息素选择下一个节点，直到达到适当的解决方案。在蚁群算法中，信息素信息分布的更新过程是基于当前迭代中的所有搜索行程，可以理解为蚂蚁的 CI 的涌现。在这个意义上，ACO 算法成功地应用于多个问题，如旅行商问题（TSP）[15,16]、数据挖掘和比例 – 积分 – 微分（PID）控制参数的优化。此外，科学家还提出了一些有效的 CI 算法，如粒子群优化算法（PSO）[17]，它模拟了鸟类的觅食。

除了在这些优化问题中使用了 CI，从群体中学习可能是解决现实世界中大数据背景下机器学习应用挑战的一种方式。例如，用于监督学习的标签对于许多应用来说可能太昂贵甚至不可能获得 [18]。因此，研究者们开发了 CI 学习技术 [19~22] 来克服这一困难。在下一节中，将看到 CI 处理大量棋类游戏标签的能力。在本文中，笔者尝试用 CI 进化理论解决工业中的问题，比如智能机器人的应用，并取得了初步的验证效果。笔者的工作有可能促进计算机科学领域对 CI 的研究，也为 CI 与深度学习和强化学习的结合铺平道路。

三、AlphaZero 中群智智能逻辑探索

在这一节中，将回顾 AlphaZero 中的理论，也涉及它之前的版本 AlphaGo Fan [23]，AlphaGo Lee [24]，AlphaGo Master [24] 和 AlphaGo Zero [24]。然后将从 CI 的角度分析这些理论。这些理论分为

两部分：①用深度神经网络代表个体；②通过强化学习使个体进化。注意笔者会将 MCTS 的细节留到下一节来做重点分析，因为 CI 是在 MCTS 中涌现的。最后将应用 AlphaZero 到一种新的游戏，即五子棋，来展现 AlphaZero 的群智算法的逻辑。

（一）AlphaZero 核心概念回顾

从实际对弈的角度来看，AlphaZero 使用 MCTS 算法进行搜索寻找最佳落子。由于搜索时间有限，不可能穷尽所有的可能落子，所以使用了策略网络来减小搜索宽度，使用了价值网络来减小搜索深度。策略网络作为采样的先验概率，以更大的概率去搜索那些可能赢棋的落子。价值网络作为状态的评价函数，不需要模拟到棋局结束便可给出胜负的预测。

从训练的角度来看，策略网络和价值网络是用强化学习的策略迭代算法训练出来的。MCTS 相当于是策略提高算子，因为搜索概率比策略网络的概率要好，用搜索概率作为标签来训练策略网络。基于 MCTS 的自我对弈相当于是策略评价算子，这里的策略指的是 MCTS 的搜索概率，因为评价的是使用搜索概率下棋的胜负，这个胜负作为标签训练价值网络。下面将换一个角度，从 CI 的角度来重新分析 AlphaZero。

（二）用深度神经网络代表个体

个体的表达能力限制了它们的智能程度。如果个体的表达能力较低，即使 CI 涌现出来，CI 也不能被个体继承。在 AlphaZero 中，个体是通过深度神经网络来代表，就是为了提高个体的表达能力。

在 AlphaGo 中，给定当前棋盘状态，策略网络用来提供下一步落子的概率分布，价值网络用来提供赢棋的概率。在 AlphaGo Fan 中策略和价值是两个独立的神经网络，每个网络有 13 个卷积层。然后在 AlphaGo Lee 中，每个卷积层的卷积核数量由 192 增加为 256。从 AlphaGo Master 到 AlphaZero，策略和价值网络被结合到一个网络当中，并且卷积层的数量增加到 39 或 79。表 1 为 AlphaGo 所有版本神经网络结构对比。AlphaZero 的棋力比 AlphaGo Lee 明显要好。而且值得一提的是监督学习得到的 AlphaZero 神经网络也比得上 AlphaGo Lee 的棋力。这个事实体现了 AlphaZero 中神经网络的作用。

AlphaZero 神经网络表现优异的原因有许多。最首要的是网络的大小。可以看到 AlphaZero 中卷积层的数量是 AlphaGo Lee 的 3 倍，这意味着 AlphaZero 中可调参数也大致是 AlphaGo Lee 的 3 倍。这表明网络的表达能力大幅提升。用这种方式，网络能够学习到 MCTS 生成的搜索概率，也就是说个体能够继承 CI 的知识。其他原因包括：①残差块降低了训练难度；②双重网络结构使得策略和价值网络被调整到一个共同的表达方式，并且提高了计算效率。

（三）通过强化学习使个体进化

一旦个体具备了足够的表达能力，下一个问题就是怎样让它们进化。为了能让个体持续进化，就需要找到进化的方向。在 AlphaZero 中，是通过个体自己的经历来找到进化方向，即通过强化学习。这样的结果就使个体能够持续进化，最后超越了之前版本以及人类专家的棋力。

在最早的版本 AlphaGo Fan 中，策略网络是先由专家知识训练的。然后用 Reinforce 算法提高策略网络。换言之，强化的网络是通过策略网络自我对弈结果训练出来的。之后，价值网络是通过强化的策略网络自我对弈结果训练出来的。在下一个版本 AlphaGo Lee 中，价值网络是由 AlphaGo 自我对弈的结果训练出来的，而不是用策略网络自我对弈，并且这个过程反复了几次。从 AlphaGo Master 到 AlphaZero，不仅价值网络是通过 AlphaGo 自我对弈的结果训练出来的，策略网络也是由 AlphaGo 生成的搜索概率训练出来的。值得一提的是 MCTS 用

表 1　AlphaGo 神经网络结构对比

	AlphaGo Fan	AlphaGo Lee	AlphaZero
卷积层数量	13	13	39 或 79
每层的卷积核数量	192	256	256

来生成搜索概率并落子。

从 AlphaGo 的发展可以总结出强化学习是进化的关键，并且自己生成的标签质量决定了进化的程度。对于价值网络，对比 AlphaGo Fan 和之后的版本，主要区别是价值网络的标签。在之后的版本里，标签更为准确，因为它们是由使用了 MCTS 落子的 AlphaGo 生成的，而不是仅用强化的策略网络。对于策略网络，从 AlphaGo Master 到 AlphaZero，是由 MCTS 生成的搜索概率作为标签，而不是由对弈结果指引的策略网络自己的落子，具体的比较总结，如表 2 所示。

之所以 MCTS 生成的标签比策略网络好是因为：MCTS 包含了多次模拟来落子，在每次模拟中，策略网络用来给出先验概率，价值网络用来更新行动价值。可以把每次模拟中的策略和价值网络当作一个个体，那么搜索概率会随着个体数量的增加而变得准确。因此，MCTS 可以提供 CI，在这里指搜索概率以及使用搜索概率下棋得到的胜负结果。在文献 [24] 中，基于 MCTS 的自我对弈被视为强化学习中的策略评价算子，但它的策略指的是 MCTS 的搜索概率，并不是原本的策略网络，与原本的策略迭代算法不完全一致。所以，更合适的观点是将 MCTS 视为 CI 算法，关于 MCTS 的更多信息将在下一节介绍。

（四）应用于五子棋时的训练结果

为了展示 AlphaZero 的群智智能逻辑，笔者将这一技术应用于一个新的游戏，即五子棋，同时也应用于五子棋的一个变体，即有禁手五子棋。训练结果将在下文展示。注意笔者对 AlphaZero 做了一些改进使得它能适应五子棋和有禁手五子棋的规则。

图 1 表示的是改进的 AlphaZero 在五子棋中的训练结果。图 1(a) 展示了改进的 AlphaZero 的棋力。注意在五子棋上同样实现了 AlphaGo Fan，它的棋力也被作为对比对象。Elo 评分是用不同选手在多样的开局下比赛算出来的，每步使用 1 s 的思考时间。对于 AlphaZero，使用一个图形处理器（GPU）来计算神经网络。图 1(b) 展示了训练时每一代神经网络在测试集上的预测准确率。准确率测量了神经网络给出的最高概率的落子的准确性。图 1(c) 展示的是训练时每一代神经网络预测测试集对弈结果的均方差（MSE）。同样的，改进的 AlphaZero 在有禁手五子棋上的训练结果，如图 2 所示。

可以看出，AlphaZero 的棋力比传统的通过专家知识构造的引擎要强。策略和价值网络从它们自身的经历中逐渐学到了自己的战术。同时也展示了 AlphaZero 可被用于不同规则的游戏。AlphaZero 的通用性是继承于表达方式的通用性，即深度神经网络，也继承于进化方法的通用性，即强化学习。并且，由 MCTS 生成的标签为强化学习提供了进化的方向。在下一节，CI 将被用来解释 MCTS 的原理。

四、ACO 和 MCTS 的比较分析

MCTS 是一种高效的启发式决策搜索算法，广泛应用于博弈游戏中。笔者就以群体算法中最具有代表性的 ACO 为例，和 MCTS 算法进行对比，并将它们应用到 TSP 问题中。然后通过应用的结果，来分析 ACO 和 MCTS 算法的共性特征。

（一）TSP 旅行商问题

TSP 问题是一个经典的组合优化问题，有下列具体描述 [25]：

$V = \{a, \cdots, z\}$ 为城市集合，$A = \{(r,s):r,s \in V\}$ 是城市中两两城市的连接的边，每个边是城市之间的距离：$\delta(r,s) = \delta(s,r)$，$(r,s) \in A$。TSP 问题是找到能够不重复访问所有城市的最短路径。在该问题中，每个城市由 $r \in V$ 都有具体的坐标值 (x_r, y_r)，因此

表 2　标签来源对比

	AlphaGo Fan	AlphaGo Lee	AlphaZero
策略的标签	监督的策略	监督的策略	AlphaGo
价值的标签	强化的策略	AlphaGo	AlphaGo

注：标签来源对比，分别是从监督学习的策略网络、强化的策略网络和使用 MCTS 的 AlphaGo 而来。

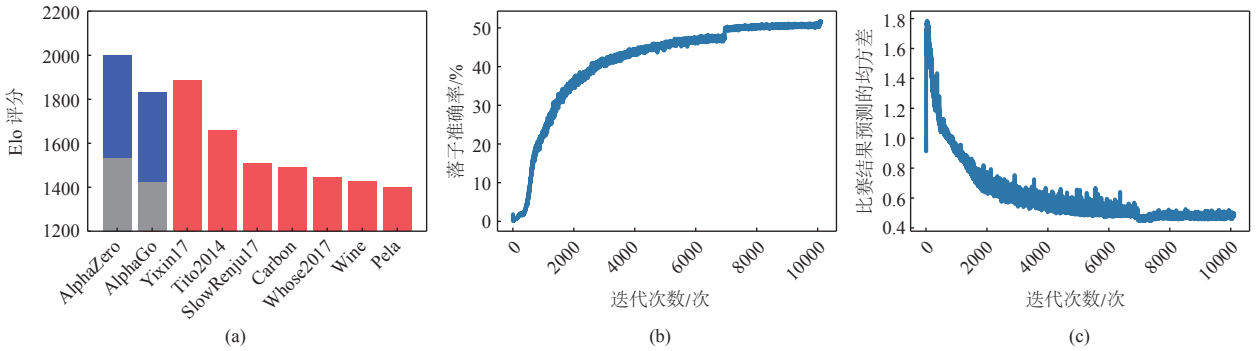

图 1　改进的 AlphaZero 在五子棋中的训练结果

注：(a) 改进的 AlphaZero 在五子棋的棋力，其中相应的策略网络的棋力用灰色表示；(b) 在测试集上的预测准确率；(c) 预测测试集对弈结果的 MSE。

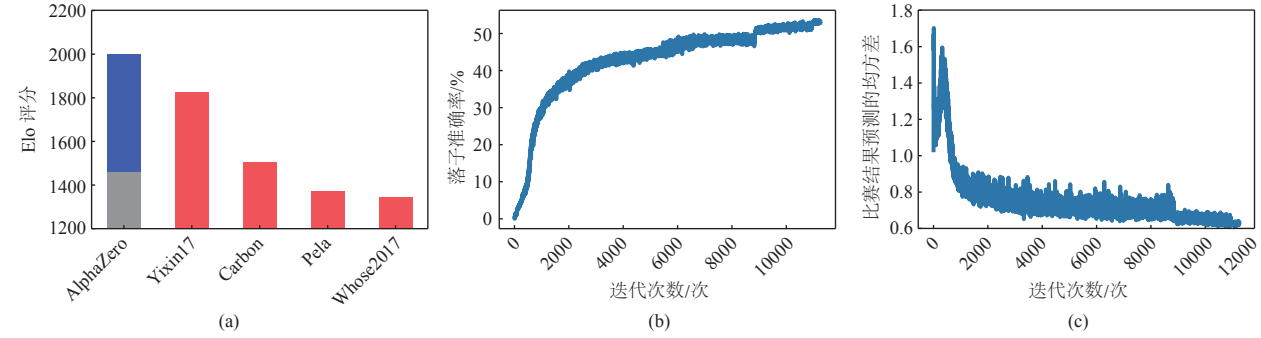

图 2　改进的 AlphaZero 在有禁手五子棋中的训练结果

注：(a) 改进的 AlphaZero 在有禁手五子棋的棋力，其中相应的策略网络的棋力用灰色表示；(b) 在测试集上的预测准确率；(c) 预测测试集对弈结果的 MSE。

也被称为欧拉形式的 TSP 问题。

TSP 问题也是非确定多项式（NP）问题的代表问题，计算复杂度与城市数量呈指数关系。

（二）ACO 蚁群算法

ACO[25~27] 算法采用了模拟真实自然环境中蚁群的行为，很好地解决了如 TSP 等组合优化问题。蚁群在搜索食物时，最开始的时候在它们的巢穴周边进行随机策略搜索，一旦有蚂蚁发现了食物，它们就把食物从食物源搬回巢穴。在搬运食物的过程中，蚂蚁会在返程的路径上释放化学信息素，信息素释放的数量取决于找到的食物的数量和质量。当之后的蚂蚁进行搜索时，能够依据信息素的多少，判断食物源的方向，更快地找到食物。蚁群通过信息素实现了多个个体的信息共享，这使得它们可以很快地找到从巢穴到食物源的最短路径。

当解决 TSP 问题时，每个迭代步由以下两个主要的步骤组成：

模拟：每只蚂蚁依据状态转移概率矩阵，按照概率分布完成一次完整的搜索，选择每一条路径的概率正比于状态转移概率矩阵

$$a_t \sim p_k(r,s)=\begin{cases}\dfrac{[\tau(r,s)]\cdot[\eta(r,s)]^{\beta}}{\sum_{u\in J_k(r)}[\tau(r,u)]\cdot[\eta(r,u)]^{\beta}}, & \text{if } s\in J_k(r)\\ 0, & \text{otherwise}\end{cases} \tag{1}$$

式（1）中，τ 为信息素；$\eta=1/\delta(r,s)$ 为路径 $\delta(r,s)$ 的倒数；$J_k(r)$ 为第 k 只蚂蚁从搜索过程中的城市 r 出发剩余需要访问的城市；β 为访问状态转移先验概率的一个超参数。

更新：一旦所有蚂蚁完成了它们的搜索，需要进行一次全局的信息素更新

$$\tau(r,s)\leftarrow(1-\alpha)\cdot\tau(r,s)+\sum_{k=1}^{m}\Delta\tau_k(r,s) \tag{2}$$

其中，

$$\Delta\tau_k(r,s)=\begin{cases}\dfrac{Q}{L_k}, & \text{if } (r,s)\in \text{tourdonebyantk}\\ 0, & \text{otherwise}\end{cases} \tag{3}$$

式（2）和式（3）中，α 为信息素衰减因子；L_k 为蚂蚁 k 途经路径的长度；m 为蚂蚁的总数量；Q 为信息素的权重因子，它决定了探索和利用的比重。

整个搜索过程由以上步骤进行迭代，直到达到

终止状态，在本文中，超参数取 $Q = 1.0$，$\alpha = 0.1$，$\beta = 1.0$。

（三）MCTS 蒙特卡洛树搜索

MCTS[28~30] 是一种能够在给定环境找到最优策略的启发式的树搜索方法。MCTS 在计算机围棋领域取得了巨大的成功，其中以 AlphaGo [23] 和 AlphaGo Zero [24] 为代表，结合了 MCTS 和深度神经网络，并使用了自我对弈强化学习实现进化，最终实现了超越人类顶尖棋手的棋力水平。

MCTS 在整个树搜索空间中，采用随机的策略进行大量模拟来评估状态价值。随着模拟的次数增加，搜索树也增加得更大，对状态价值的估计也更加准确。进行树搜索的策略在搜索过程中也在不断改进，渐渐地，树搜索策略收敛于最优策略，状态价值估计也收敛于真实的状态价值。

图 3(a) 表示了 MCTS 搜索中的一个迭代步中的四个步骤 [28]，具体步骤如下：

选择（selection）：从树的根节点开始，依照选择策略递归进行子节点选取，直到达到搜索树的叶节点。在 TSP 问题中的树搜索策略是在所有子节点中根据置信上界方法选取（UCT）。

$$
\begin{aligned}
a_t &= \operatorname{argmax}(Q(s,a) + u(s,a)) \\
Q(s,a) &= \frac{\overline{L}}{L(s,a)/N(s,a)} \\
u(s,a) &= C_p P(s,a) \frac{\sqrt{\sum_b N(s,a)}}{1 + N(s,a)}
\end{aligned}
\tag{4}
$$

式（4）中，s 为当前节点状态；$L(s,a)$ 为经过边 (s,a) 的总路径长度；$N(s,a)$ 为边 (s,a) 被访问过的总次数；L 为被访问过所有合法路径的长度的平均值；C_T 为 UCT 方法中的一个超参数，用来平衡探索和利用。搜索的先验概率为 $P(s,a)$，为了和 ACO 算法中的先验概率保持一致，取边长度 $\delta(s,a)$ 的倒数

$$
P(s,a) = \frac{1/\delta(s,a)}{\sum_b 1/\delta(s,a)} \tag{5}
$$

扩展（expansion）：对树叶节点进行扩展，选

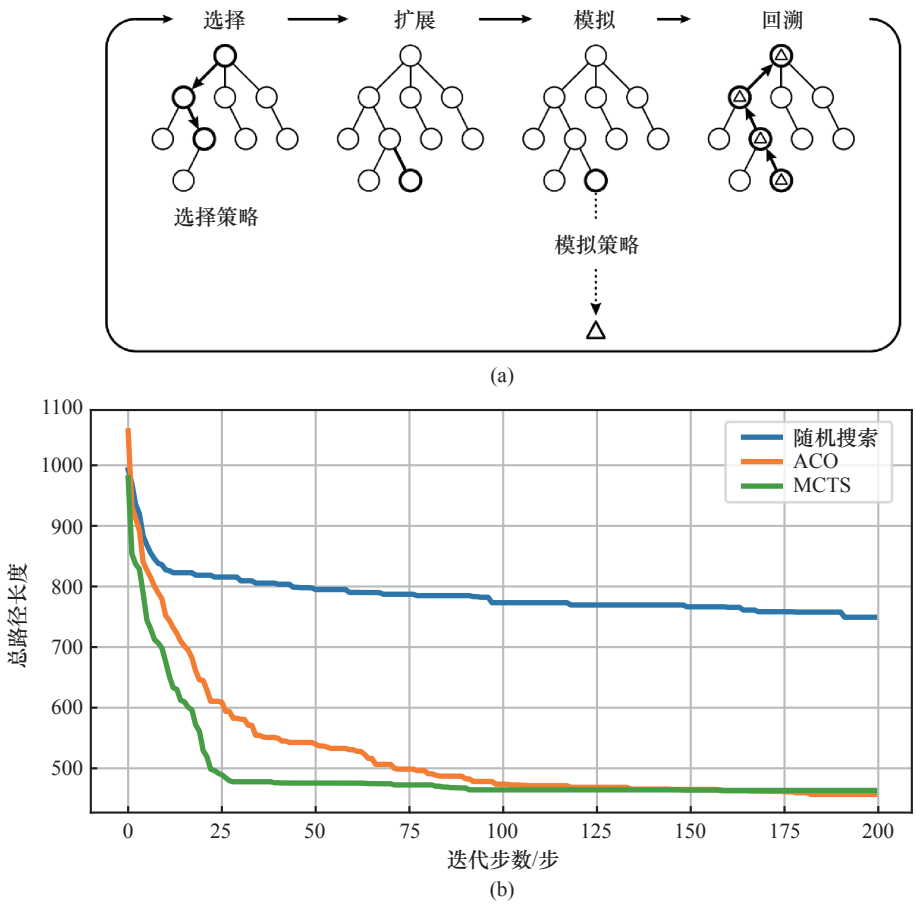

图 3　MCTS 每个迭代步的四个步骤（a）及 ACO、MCTS 和随机搜索收敛曲线（b）

取当前节点之后所有的可行城市作为当前节点的子节点。

模拟（simulation）：当到达叶节点后，按照默认策略行走直到达到终点，得到当前路径长度 l_t。模拟的默认策略是按照正比于先验概率 $P(s,a)$ 选择当前节点的可行城市。

回溯（backpropagation）：完成一次模拟之后，按照当前的模拟结果对整个搜索树进行更新。

$$N(s,a) \leftarrow N(s,a)+1,\ L(s,a) \leftarrow L(s,a)+l_t \quad (6)$$

当经过了指定次数的迭代，最终依照访问次数最多的城市进行选择。在本文中，超参数 $C_p=3.0$。

（四）结果与分析

由于欧拉 TSP 问题中的城市间的连接距离是按照实际坐标点距离计算，因此是一个无向图问题，并且搜索路径是一个闭环，因此整个搜索图也可以被视作一个树搜索结构。为了和 MCTS 对比，ACO 每次都从一个固定的城市出发进行搜索，固定的城市就是 MCTS 中的搜索起始根节点。两种算法的详细配置见表 3。

本文将这两种算法应用于 30 个城市的 TSP 问题，另外为了与这两种方法做对照，加入了纯随机搜索作为对照。使用这三种方法分别进行了 10 次 TSP 问题优化，最终结果如图 3(b) 和表 4 所示。

相比于随机搜索，ACO 和 MCTS 都体现了良好的收敛性，在前 100 迭代步中，MCTS 略微优于 ACO，但是在后半程出现了搜索停滞。一个主要原因是由于 MCTS 搜索为一个树状结构，而 ACO 搜索为一个网状结构，ACO 对于局部区域路径优化的能力更强。

对比 ACO 和 MCTS 中的每一个迭代步中的具体算法可以发现，MCTS 具有和 ACO 相似的机理，在每一个迭代步中，每个个体需要按照特定策略进行搜索，并依据全局群体共享信息实时更新策略。这两种算法相似点有以下几点：

模拟策略：在 ACO 中，进行模拟的策略是按照状态转移概率矩阵得到，在 MCTS 中，搜索树中的部分是依照 UCT 策略得到，模拟的部分采用默认模拟策略。

群体信息共享：在 ACO 中，所有的输出结果都更新到全局信息素中，全局信息素决定了状态转移概率矩阵。在 MCTS 中，模拟的结果更新到 $Q(r,s)$，这影响到了下一次在搜索树中选择的 UCT 策略。

平衡探索和利用：在 ACO 中，模拟的行动选择正比于概率分布，同时保证了探索和利用，受超参数 Q 影响。在 MCTS 中，UCT 算法保证了平衡探索和利用，受超参数 C_p 影响。

这些特征同样也是群智算法的关键特征。从实验结果可以看到，虽然 MCTS 算法没有显式的群体搜索的概念，其搜索的机理体现了群智涌现

表 3　ACO、MCTS 的算法超参数设置

	ACO	MCTS
搜索方式	固定一点作为起始根节点	蚁群数量为 1，固定起点
先验概率	状态转移先验概率 η^{β} $\eta=1/\delta(r,s),\beta=1.0$	选择先验概率 $P(s,a)$ $P(s,a)=\frac{1/\delta(s,a)}{\sum_b 1/\delta(s,a)}$
其他超参数	信息素权重因子 $Q=1.0$ 信息素衰减因子 $\alpha=0.1$	UCT 选择权重 $C_p=3.0$

表 4　ACO、MCTS 和随机搜索结果

	最优值	平均值
ACO	426.75	456.97
MCTS	450.74	463.51
随机搜索	694.79	749.25

的特征，因此可以被视作群智算法。群智涌现是保证 ACO 和 MCTS 具有良好搜索收敛性的关键机制。

五、群智进化理论

在深入研究了 AlphaZero 程序和 MCTS 算法之后，其下隐藏的智能进化机制被完整地发现了。AlphaZero 的成功主要取决于两个因素，一个是使用深度卷积神经网络来表示个体智能，另一个是使用 MCTS 使 CI 涌现并高于个体智能。深度卷积神经网络能够通过用合适的目标标签训练来进化其智能。MCTS 算法能够通过 CI 涌现来生成合适的目标标签。在强化学习环境中结合这两个因素，个体智能进化的正反馈就形成了。

因此，笔者提出了一个 CI 进化理论，并将其作为走向 AGI 的通用框架。第一，定义一个深度神经网络来表示个体智能；第二，使用 CI 算法使 CI 涌现并高于个体智能；第三，利用这个更高的 CI 进化个体智能。最后，在强化学习环境中不断重复涌现 – 进化的步骤，以形成个体智能进化的正反馈，直到智能收敛。通用 AGI 进化框架流程图，如图 4 所示。

用 $p(k)$ 和 $v_p(k)$ 表示第 k 次迭代中的个体策略和个体状态价值，其中 $p(k)$ 由深度神经网络来表达，$v_p(k)$ 是衡量个体智能程度的标准，可以通过 $p(k)$ 与环境交互得到（例如在围棋中将与几个对手引擎下足够多盘棋作为环境，下赢的奖励为 1，下输的奖励为 0，那么 $v_p(k)$ 就等于策略 $p(k)$ 的胜率，AlphaZero 中使用 Elo 评分衡量个体智能程度，本质也是先通过与环境交互的胜率计算得来，

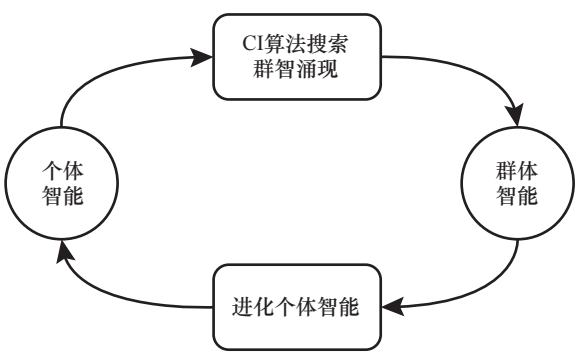

图 4　通用 AGI 进化框架流程图

再通过个体 Elo 与环境平均的 Elo 之差便可反推胜率）；用 $\pi(p(k))$ 和 $v_\pi(v_p(k))$ 表示群体策略和群体状态价值，其中 $\pi(p(k))$ 由 CI 算法产生，$v_\pi(v_p(k))$ 通过 $\pi(p(k))$ 与环境交互得到；用 v_* 表示最优状态价值，通常有 $v_p(k) \leqslant v_\pi(v_p(k)) \leqslant v_*$；用 $\alpha(k) \in [0,1]$ 表示个体智能学习 CI 的程度，即在 $v_p(k)$ 和 $v_\pi(v_p(k))$ 之间做线性插值；用 $\beta(k) = v_\pi(v_p(k)) - v_p(k) \in [0, v_* - v_p(k)]$ 表示 CI 高于个体智能的程度。如果将 $v_p(k)$ 视为动力系统的状态量，将 $v_\pi(v_p(k))$ 视为动力系统的控制量，这种正反馈可以表示成离散时间系统

$$\begin{aligned} v_p(k+1) &= (1-\alpha(k))v_p(k) + \alpha(k)v_\pi(v_p(k)) \\ &= v_p(k) + \alpha(k)(v_\pi(v_p(k)) - v_p(k)) \\ &= v_p(k) + \alpha(k)\beta(k) \\ &= v_p(0) + \sum_{k=0}^{k} \alpha(k)\beta(k) \end{aligned} \tag{7}$$

目标是个体状态价值达到最优，即 $\lim\limits_{k\to+\infty} v_p(k) = \lim\limits_{k\to+\infty} v_\pi(v_p(k)) = v_*$，理想情况是在达到最优前的任意时刻 k，有 $\alpha(k) > 0$ 和 $\beta(k) > 0$，即 $v_p(k)$ 单调递增，且 $v_p(0) + \sum_{k=0}^{+\infty} \alpha(k)\beta(k) = v_*$。当然，实际应用中也可能情况异常，存在某些时刻 k，$\alpha(k) < 0$ 或 $\beta(k) > 0$，导致正反馈中断。为了保证正反馈的持续进行，需要有理论的支撑，并且在实际应用中调节超参数来弥补理论和实际的间隙。

其中，$\alpha(k) > 0$ 由神经网络的训练来保证，例如使用损失函数 $l = -\pi^{\mathrm{T}}(p(k))\log p(k)$ 和梯度下降来训练神经网络。根据 Gibbs 不等式 [31]，当且仅当 $p(k) = \pi(p(k))$ 时，l 达到最小值。虽然有理论保证，但 $\alpha(k)$ 受神经网络的结构和梯度下降算法中的超参数影响，不一定能达到 $p(k) = \pi(p(k))$，即 $\alpha(k) = 1$。实际应用中需要合理调节这些超参数使得 $\alpha(k) > 0$ 即可。

另一方面，$\beta(k) > 0$ 由 CI 算法来保证。在最早的蚁群算法 ant system（AS）[27] 的基础上改进后，很多蚁群算法的扩展都有了收敛性的保证，graph-based ant system（GBAS）算法能收敛到最优行动的概率为 1 [32]，而常用的 ant colony system（ACS）[25] 和 max-min ant system（MMAS）[12] 算法能收敛到最优行动的概率大于一个下界值 [33]。MCTS 从最初的版本改进到 UCT，也就是将置信上界（UCB）[34] 加入到选择中，能收敛到最优行动的概率为 1[30]。AlphaZero 是将

predictor UCB（PUCB）算法加入到 MCTS，而单独的 PUCB 算法能收敛到最优行动的概率大于一个下界值 [35]。虽然 AlphaZero 中的 MCTS 没有理论证明，但从应用效果来看也可以使得 $\beta(k)>0$，实际应用中需要合理调节超参数来弥补理论和实际的间隙。

在完美智能 v_* 有限的情况下，CI 进化有两种类型的智能收敛。一种是个体智能收敛到一个和 CI 相同的极限。这意味着或者是完美的智能已经到达，即 $\lim_{k\to+\infty}\beta(k)=0$，$\lim_{k\to+\infty}v_p(k)=\lim_{k\to+\infty}v_\pi(v_p(k))=v_*$，或者是 CI 算法不足以使得更高的群体智能涌现，即 $\lim_{k\to+\infty}\beta(k)=0$，$\lim_{k\to+\infty}v_p(k)=\lim_{k\to+\infty}v_\pi(v_p(k))<v_*$。另一种是个体智能收敛到一个低于 CI 的极限，这意味着或者是个人智能的容量不够大，或者是训练方法不再有效，即 $\lim_{k\to+\infty}\alpha(k)=0$ 且 $\lim_{k\to+\infty}v_p(k)=v_p(0)+\sum_{k=0}^{+\infty}\alpha(k)\beta(k)<\lim_{k\to+\infty}v_\pi(v_p(k))\leqslant v_*$。

与现有的机器学习方法相比，CI 进化理论具有一定的优势。深度学习是强大的，但依赖于大量高质量的标签数据过于昂贵。强化学习通过廉价的奖励信号为个体智能提供了进化环境，但由于试错性质，学习效率较低。CI 算法能够使 CI 从无到有，但缺乏一种进化个体智能的机制。CI 进化理论结合深度学习、强化学习和 CI 算法的优势，通过 CI 的涌现，使个体智能高效、低成本地进化。这种进化可以从零开始，因此 CI 进化理论是向 AGI 迈进的一步。

六、智能机器人应用

传统的机器人可以利用一些计算机视觉或专家系统技术来实现某种智能行为，但它们缺乏学习或进化能力来自动适应环境变化。例如，焊接机器人能够通过 3D 视觉系统和基于传统特征的视觉算法来跟踪焊缝。但是，为了使焊接机器人工作正常，必须在新的焊接环境中手动调整一些关键参数。因此，机器人工业迫切需要能够像人类一样自动适应环境的智能机器人。

CI 进化理论在智能机器人中有着天然的应用，它通过传感器、智能体和执行器的闭环提供了一个强化学习环境。该理论的应用称为智能模型。为了促进智能模型的实现，一个云端平台被开发出来帮助创建和进化智能机器人的智能模型。

面向工业应用的智能模型主要分为三类，视觉检测、数据预测、参数优化，其中参数优化具有最广泛的需求。作为这一概念的验证，一个焊接机器人的焊接参数优化智能模型已经在云端平台上实现。

随着科技的发展，在钢铁材料的焊接领域，机器人焊接逐渐取代了以往的人工焊接。在焊接机器人实施焊接过程中，控制焊接的参数会直接影响焊接的质量。焊接的参数有焊枪移动速度、电流、电压、焊枪角度等。焊接参数需要焊接工程师根据焊接板材材质、焊缝宽度以及焊接板材厚度等场景手工调节优化焊接参数，为满足焊接机器人在工业应用中智能化的需求，提出用深度学习和强化学习的技术，结合焊接机器人 3D 视觉系统，实现焊接机器人根据焊接场景的不同实现焊接参数的自动调节，或者说实现由焊接场景到焊接参数最优的映射关系。

考虑最简单的焊接场景，输入特征只保留一个焊缝宽度，从零开始均匀增加，输出参数仅控制焊枪移动速度。

焊接参数优化的目标是得到最好的焊接质量，具体来说，就是对于较小的焊缝宽度，希望焊接后的焊料宽度保持在 5 mm；对于较大的焊缝宽度，希望焊料宽度比焊缝宽度大 2 mm，并且不论焊缝多宽，理想的焊料高度都是 1 mm。图 5 为焊缝宽度和焊接板材长度的关系曲线，图 6 为理想的焊料宽度和焊缝宽度的关系曲线。

在一条焊缝的焊接过程中从起始点开始把每隔等长的一小段间距作为一个焊接点，焊接点的个数

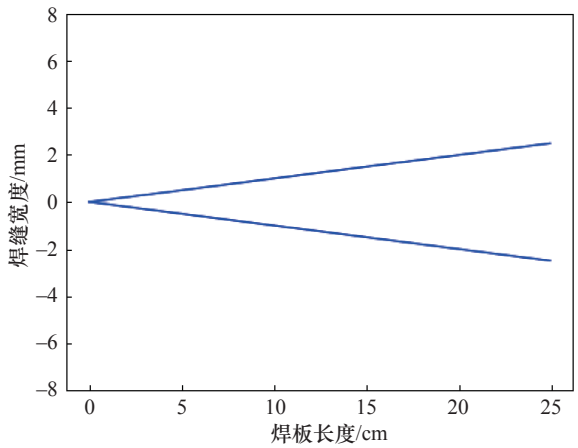

图 5　焊缝宽度和焊接板材长度的关系曲线

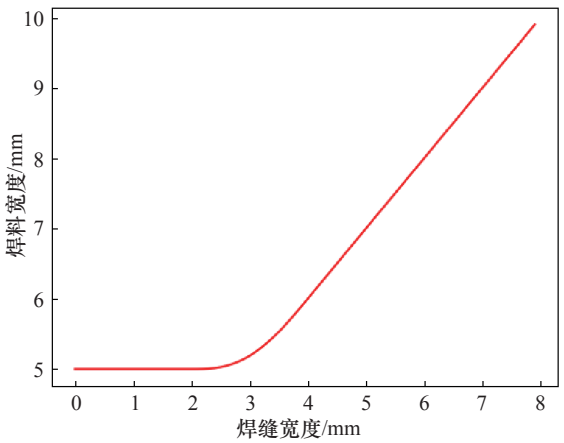

图 6　理想的焊料宽度和焊缝宽度的关系曲线

用 n 表示，每一时刻未焊接点的焊缝宽度和已焊接点的焊料宽度和高度分别用 g_i，w_i，h_i 表示，焊接到第 i 个焊接点的时刻用 t_i 表示。笔者定义一个简化的马尔可夫决策过程（MDP）模型，假设 t 时刻的环境状态 $s_t=g_i$，t 时刻智能体的行动就是焊枪在第 i 个焊点的移动速度 v_{t_i}，即 $a_t=v_{t_i}$，并且假设折扣因子为 0，即仅考虑即时奖励，把每一焊接点的实际焊接效果和理想焊接效果之间的偏差作为这一时刻的奖励。

图 7 为焊接参数优化智能模型的训练流程图。

为了训练这个智能模型，首先到焊接现场采集实际焊接效果数据，然后离线训练价值网络，最后用这个训练价值网络训练策略网络，即焊接智能体。图 8 为策略网络焊枪移动速度与焊缝宽度的关系曲线。

在云端平台上部署了这个智能模型，并到焊接现场进行了测试验证（见图 9），实现了较好的焊接质量。可以看出，针对线性变宽的直焊缝，得到的训练策略网络基本符合要求。

对于简单的焊接场景，单个智能体离线强化学习就可以达到较高的智能水平，即焊接质量。如果是复杂的焊接场景，就需要先实现在线焊接质量评估，然后根据群智进化理论进行在线智能进化，才能够实现更高的智能水平。

七、结语

CI 涌现和深度神经网络进化是 AlphaZero 程序在很多游戏中达到超人性能的关键因素。将 CI 与深度学习和强化学习相结合，就得出了 CI 进化理论。并对该理论在焊接机器人中的示范应用进行了讨论。这一理论是走向 AGI 的通用框架，因此期待在未来有越来越多的应用和进一步的理论探讨。

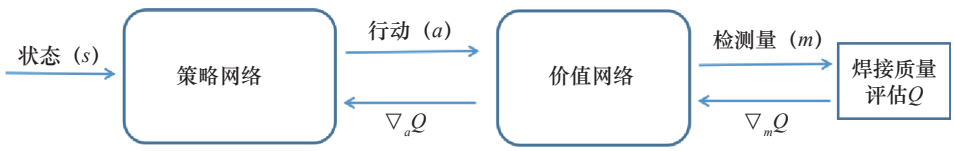

图 7　焊接参数优化智能模型的训练流程图

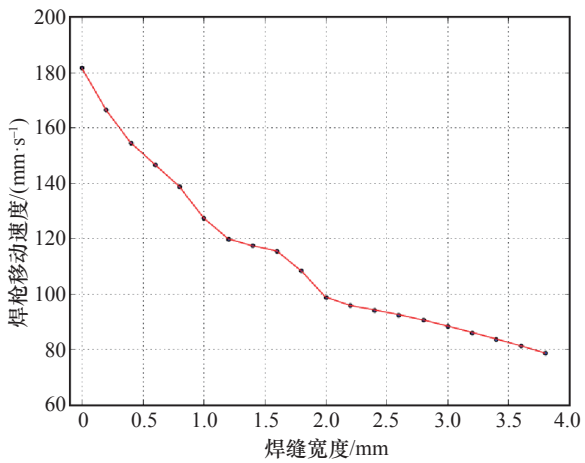

图 8　策略网络焊枪移动速度与焊缝宽度的关系曲线

图 9　焊接现场测试验证

参考文献

[1] Landemore H. Democratic reason: Politics, collective intelligence, and the rule of the many [M]. Princeton: Princeton University Press, 2012.

[2] Wolpert D H, Tumer K, Frank J. Using collective intelligence to route internet traffic [M]. Cambridge: MIT Press, 1999.

[3] Wolpert D H, Tumer K. Collective intelligence, data routing and braess'paradox [J]. Journal of Artificial Intelligence Research, 2002, 16(4): 708–714.

[4] Tumer K, Wolpert D H. Collectives and the design of complex systems [M]. Berlin: Springer-Verlag, 2004.

[5] Ng A Y, Harada D, Russell S J. Policy invariance under reward transformations: Theory and application to reward shaping [C]. San Francisco: ICML'99 Proceedings of the Sixteenth International Conference on Machine Learning, 1999.

[6] Marden J R, Shamma J S. Game theoretic learning in distributed control—Handbook of dynamic game theory [M]. Berlin: Springer International Publishing, 2017.

[7] Samuel A L. Some studies in machine learning using the game of checkers II—Recent progress [J]. IBM Journal of Research and Development, 1967, 11: 601–617.

[8] Bon G L. The crowd: A study of the popular mind [M]. Berlin: Springer-Verlag, 2009.

[9] Thomas R L, Malone W, Dellarocas C. The collective intelligence genome [J]. IEEE Engineering Management Review, 2010, 55(1): 21–31.

[10] Woolley A W, Chabris C F, Pentland A, et al. Evidence for a collective intelligence factor in the performance of human groups [J]. Science, 2010, 330(6004): 686–688.

[11] Colorni A, Dorigo M, Maniezzo, et al. Distributed optimization by ant colonies [C]. Berlin: The 1st European Conference on Artificial Life, 1992.

[12] Stutzle T, Hoos H H. Max-min ant system [J]. Future Generation Computer Systems, 2000,16(8): 889–914.

[13] Zlochin M, Birattari M, Meuleau N, et al. Model-based search for combinatorial optimization: A critical survey [J]. Annals of Operations Research, 2004, 131(1–4): 373–395.

[14] Dorigo M, Birattari M, Stutzle T. Ant colony optimization [J]. IEEE Computational Intelligence Magazine, 2006, 1(1): 28–39.

[15] Rego C, Gamboa D, Glover F, et al. Traveling salesman problem heuristics: Leading methods, implementations and latest advances [J]. European Journal of Operational Research, 2011, 211(3): 427–441.

[16] Rabiner L R. Combinatorial optimization: Algorithms and complexity [J]. IEEE Transactions on Acoustics, Speech, and Signal Processing, 1984, 32(6): 1258–1259.

[17] Poli R, Kennedy J, Blackwell T. Particle swarm optimization an overview [J]. Swarm Intelligence, 2007, 1(1): 33–57.

[18] Rodrigues F, Pereira F C, Ribeiro B. Learning from multiple annotators: Distinguishing good from random labelers [J]. Pattern Recognition Letters, 2013, 34(12): 1428–1436.

[19] Yan Y, Fung G, Rosales R M, et al. Active learning from crowds [C]. Bellevue: The 28th International Conference on Machine Learning, 2011.

[20] Long C, Hua G, Kapoor A. Active visual recognition with expertise estimation in crowd sourcing [C]. Sydney: The IEEE International Conference on Computer Vision, 2013.

[21] Zhao Z, Yan D, Ng W, et al. A transfer learning based framework of crowd-selection on twitter [C]. Birmingham: The 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2013.

[22] Fang M, Yin J, Zhu X. Knowledge transfer for multi-labeler active learning [C]. Prague: The Joint European Conference on Machine Learning and Knowledge Discovery in Databases, 2013.

[23] Silver D, Huang A, Maddison C J, et al. Mastering the game of Go with deep neural networks and tree search [J]. Nature, 2016, 529(7587): 484–489.

[24] Silver D, Schrittwieser J, Simonyan K, et al. Mastering the game of Go without human knowledge [J]. Nature, 2017, 550(7676): 354–359.

[25] Dorigo M, Gambardella L M. Ant colony system: A cooperative learning approach to the traveling salesman problem [J]. IEEE Transactions on evolutionary computation, 1997, 1(1): 53–66.

[26] Dorigo M, Blum C. Ant colony optimization theory: A survey [J]. Theoretical Computer Science, 2005, 344(3): 243–278.

[27] Dorigo M, Maniezzo V, Colorni A. The ant system: Optimization by a colony of cooperating agents [J]. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 1996, 26(1): 29–41.

[28] Browne C B, Powley E, Whitehouse D, et al. A survey of Monte Carlo tree search methods [J]. IEEE Transactions on Computational Intelligence and AI in games, 2012, 4(1): 1–43.

[29] Coulom R. Efficient selectivity and backup operators in Monte-Carlo tree search [C]. Turin: International Conference on Computers and Games, 2006.

[30] Kocsis L, Szepesvári C. Bandit based Monte-Carlo planning [C]. Berlin: European Conference on Machine Learning, 2006.

[31] Brémaud P. An introduction to probabilistic modeling [M]. Berlin: Springer Science & Business Media, 2012.

[32] Gutjahr W J. A graph-based ant system and its convergence [J]. Future Generation Computer Systems, 2000, 16(8): 873–888.

[33] Stutzle T, Dorigo M. A short convergence proof for a class of ant colony optimization algorithms [J]. IEEE Transactions on Evolutionary Computation, 2002, 6(4): 358–365.

[34] Auer P, Cesa-Bianchi N, Fischer P. Finite-time analysis of the multiarmed bandit problem [J]. Machine Learning, 2002, 47(2–3): 235–256.

[35] Rosin C D. Multi-armed bandits with episode context [J]. Annals of Mathematics and Artificial Intelligence, 2011, 61(3): 203–230.