适应度二次选择的 QPSO 和 SA 协同搜索大规模离散优化算法

张兆娟[1]，王万良[1]，唐继军[2]

（1. 浙江工业大学计算机科学与技术学院，浙江 杭州 310023；2. 天津大学智能与计算学部，天津 300050）

摘 要：针对大规模离散工程优化问题，提出一种改进的离散量子粒子群优化算法（IDQPSO-SA）。首先，提出一种适应度的二次选择更新平均最优位置策略，使 QPSO 能够适用离散空间的优化问题。其次，引入二次切割与连接（DCJ）排序策略加速搜索进程。最后，在 QPSO 并行搜索基础上，引进模拟退火（SA）的概率突跳性，协同进行全局搜索。在大规模、高维离散工程优化问题上进行了测试，并同已有算法进行比较，结果表明，IDQPSO-SA 进一步提高了面向大规模离散优化问题时的搜索效率，并有效提升了算法的性能。

关键词：协同搜索；量子粒子群；模拟退火；二次切割与连接排序；离散优化

中图分类号：TP301

文献标识码：A

doi: 10.11959/j.issn.1000−436x.2020173

Second fitness selection QPSO and SA cooperative search for large-scale discrete optimization algorithm

ZHANG Zhaojuan[1], WANG Wanliang[1], TANG Jijun[2]

1. College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou 310023, China

2. College of Intelligence and Computing, Tianjin University, Tianjin 300050, China

Abstract: To address the large-scale discrete optimization problem, a cooperative optimization algorithm called IDQPSO-SA was proposed. First, a strategy by applying two selections on the averaging fitness values to update the mean best position was presented, which could overcome the deficiency that QPSO was not applicable for discrete problems. Second, the double cut joining (DCJ) sorting strategy was incorporated into IDQPSO-SA, since the DCJ sorting strategy could considerably reduce the search space. Finally, the probability jumping ability of simulated annealing (SA) was combined with the parallel search of QPSO, and the global search was carried out collaboratively. By comparing with existing algorithms, the experimental results show that IDQPSO-SA further improves the search efficiency and has a comparable performance when faced with large-scale discrete optimization problems.

Key words: cooperative search, QPSO, SA, DCJ sorting, discrete optimization

1 引言

近年来，海量数据资源和计算能力的提升对大数据时代网络优化等问题产生巨大影响，传统优化算法难以面对大规模优化问题时搜索空间急剧增长的挑战。针对通信领域，不同的任务调度方式影响数据中心的利用率、能耗和通信网络效果。另一方面，计算智能算法在面对大规模、复杂、高维的优化问题时，优化能力会受到限制。此时，单一算法的优化能力大大削减，但将多种算法协同能够发挥混合算法的效率，从而提升优化的性能[1-2]。

面对大规模、高维数据情形，混合优化策略可用于特征选择、入侵检测、通信网络的优化调度等领域。Gheyas 等[3]提出模拟退火（SA, simulated annealing）和遗传算法（GA, genetic algorithm）的结合算法，可以基于 GA 的全局搜索和 SA 的避开

收稿日期：2020−03−20；**修回日期**：2020−06−10

基金项目：国家自然科学基金资助项目（No.61873240）

Foundation Item: The National Natural Science Foundation of China (No.61873240)

局部最优能力对大规模的特征子集进行选择。张震等[4]采用遗传算子对粒子群算法进行了改进，并联合禁忌搜索对入侵检测的高维数据特征进行选择。王晟等[5]提出一种基于遗传算法和禁忌搜索算法混合优化的移动代理测量调度方法，用于无线传感器网络中移动代理派遣次序的优化调度。叶苗等[6]结合问题实际背景设计出混合人工蜂群求解算法，对无线传感器网络中新的最小暴露路径问题进行求解。

一个优化算法的性能主要取决于以下 4 个方面的能力[7]：较好的全局搜索能力、快速收敛到最优解附近、较好的局部搜索能力、较高的计算效率。面对大规模、高维、离散搜索空间急剧增长的挑战，单一算法的优化能力呈现一定的局限性，协同优化算法能够克服上述不足。由于计算智能算法一般存在早熟现象，因此容易陷入局部最优。计算智能算法主要从初始解的选取和局部最优能力的跳出 2 个方面进行改进。

由于种群规模的存在能够增加解的多样性、SA 突跳性有助于量子粒子群优化（QPSO, quantum-behaved particle swarm optimization）算法跳出局部最优，本文提出了一种改进的离散量子粒子群和模拟退火协同优化（IDQPSO-SA, improved discrete QPSO combined with SA）算法。IDQPSO-SA 引入适应度二次选择机制，使量子粒子群优化算法适合于求解离散优化问题，而且采取镶嵌结构，结合了 SA 和 QPSO 各自的优点。IDQPSO-SA 的整个搜索包含 2 个阶段：先利用 QPSO 的并行搜索和保留历史信息能力执行搜索；再将 QPSO 搜索的个体最优解作为 SA 初始解，利用 SA 概率突跳性来提升全局搜索能力。

2　基于适应度二次选择的离散 QPSO 和 SA 协同优化算法

2.1　基于适应度二次选择的全局平均最优位置更新策略

受量子空间中粒子运动的启发，Sun 等[8]于 2004 年提出了一种能够保证全局收敛的 QPSO 算法。由于 QPSO 算法只需控制一个参数，全局搜索能力较强，已广泛应用于网络通信聚类[9]、最优设计等领域[10]。QPSO 算法中平均最优位置的引入能够提升搜索后期粒子跳出局部最优解的概率。但是，由于传统 QPSO 算法是针对连续空间设计的，平均最优位置计算方法是直接对所有粒子个体位置进行连续求和再平均而得。因此，已有 QPSO 不适合离散工程优化问题。

本文引进适应度二次选择机制，提出一种适用于离散优化问题的全局平均最优位置更新策略。首先，对所有适应度值进行平均，距离平均适应度值位置最近的个体则为初次得到的平均最优位置；然后，选择大于第一次求得的平均适应度值的个体，并和第一次求得的平均适应度值进一步进行平均，得到第二次平均的适应度值；最后，选择距离第二次求得的平均适应度值位置最近的个体，确定为最终平均最优位置。

平均最优位置更新式为

$$f(\mathrm{cbest})=\left(\frac{1}{M}\sum_{i=1}^{M}f(\mathrm{pbest}_{i1}),\ \frac{1}{M}\sum_{i=1}^{M}f(\mathrm{pbest}_{i2}),\cdots,\frac{1}{M}\sum_{i=1}^{M}f(\mathrm{pbest}_{id})\right) \tag{1}$$

其中，M 表示种群数目，$f(\mathrm{pbest}_i)$ 表示个体最优对应的适应度值，cbest 表示第一次选择得到的平均最优位置，f(cbest) 表示第一次求得的平均适应度值。

二次选择策略考虑了量子机制的不确定性，并从目标函数的角度提出了全局平均最优位置更新的方法；此外，充分考虑了种群分布带来的影响，即第一次选择将整个种群都进行了平均，并在第二次选择时保留了整个种群较优的个体。综合分析，本文提出的二次选择的更新策略有助于保留整个种群的多样性，也适用于任何离散工程优化问题。

2.2　更新进化流程

步骤 1　采用个体最优和全局最优位置的保优原则更新局部吸引子。

IDQPSO-SA 中局部吸引子的进化更新采用保优原则，通过交换序列实现迭代更新。针对离散工程优化问题，交换操作是指就维度而言进行个体之间位置的交换，多个交换操作组成了交换序列。局部吸引子更新式为

$$P_{id}=\mu\mathrm{pbest}_i\oplus(1-\mu)\mathrm{gbest} \tag{2}$$

其中，P_{id} 表示局部吸引子，μ 表示 0～1 的随机数，pbest 表示个体最优位置，gbest 表示整个种群的全局最优，符号“⊕”表示交换 pbest 和 gbest 的位置。

由于 P_{id} 的更新迭代综合考虑了整个种群的局部最优和全局最优，即 pbest 和 gbest 对整个种群的进化都有影响，从左到右依次对比 pbest 和 gbest 序

列并交换更优维度，可以使 P_{id} 向更优的方向进化。

步骤 2 采用个体和平均最优位置、局部吸引子二次保优原则更新个体位置。

IDQPSO-SA 的个体进化更新主要依据个体位置、平均最优位置、局部吸引子保优原则进行，其中个体位置和平均最优位置分别用 X_{id} 和 mbest 表示。整个更新过程包含 2 个阶段：首先依据 mbest 和 X'_{id} 从左到右交换更优维度得到一个基本交换序列 ss_1 和进化位置 X_{id}，即 $\mathrm{ss}_1=\sum_{i=1}^{n}\mathrm{so}_i=(\mathrm{so}_1,\mathrm{so}_2,\cdots,\mathrm{so}_n)$，其中 so_i 表示从 X_{id} 到 mbest 需要进行的交换算子；然后，根据式 $X_{id}(t+1)=P_{id}\oplus X'_{id}$，进一步和局部吸引子 P_{id} 进行第二次交换。IDQPSO-SA 的个体进化更新式为

$$X_{id}(t+1)=\begin{cases}P_{id}\oplus\beta\left|\mathrm{mbest}\oplus X_{id}(t)\right|\ln\left(\dfrac{1}{u}\right), & u<0.5\\ P_{id}\oplus\beta\left|X_{id}(t)\oplus\mathrm{mbest}\right|\ln\left(\dfrac{1}{u}\right), & u\geqslant 0.5\end{cases}\tag{3}$$

其中，$X_{id}(t+1)$ 表示个体当前位置，t 表示当前迭代次数，β 表示扩张–收缩因子。

步骤 3 采用 SA 嵌入量子粒子群优化算法更新个体。

SA 算法[11]主要依据不可逆动力学的思想，在某一温度下经过不断降温，在全局空间中基于蒙特卡罗（Monte Carlo）迭代启发式随机搜索最优解，同时能以一定概率跳出局部极小值并最终趋于全局最优。由于 QPSO 搜索存在进化缓慢、“早熟”、后期易陷入局部最优现象，本文结合 SA 和 QPSO 各自的优点，提出 IDQPSO-SA。

SA 是一种根据给定函数利用概率方式获取近似全局最优解的启发式算法，能够通过突跳性来扩大搜索范围和接受较差解，从而避免算法陷入局部最优。传统模拟退火算法若采取随机产生初始解的方式，适应能力较差。IDQPSO-SA 算法采用镶嵌结构，其中模拟退火进行种群更新的主要思路为：在初始阶段，将量子粒子群优化算法搜索得到的个体最优位置作为输入初始解；然后，由状态产生函数产生新个体，利用状态接受函数以一定概率接受新个体，温度按一定比例下降并将该新个体作为下一轮迭代时的当前解；不断迭代直至温度达到终止条件后产生最终解，完成整个 IDQPSO-SA 算法的混合搜索。

2.3 IDQPSO-SA 的实现

IDQPSO-SA 中，每个个体分别独立进化，每一次进化促使整个种群更好地适应整个环境。随着迭代的不断进化，搜索到的全局最优解越来越接近理论意义上的最优。当适应度函数评估次数达到终止条件时，整个搜索进程停止；否则，重复搜索直至收敛。本文提出的 IDQPSO-SA 算法的流程如图 1 所示。

3 基于 IDQPSO-SA 的祖先基因推断优化

3.1 祖先基因推断优化问题的研究现状

最近，全球疫情给全世界带来了新的挑战，而对冠状病毒基因组序列进行分析以确定溯源具有非常大的现实意义。事实上，针对新冠病毒溯源分析，即祖先基因推断属于典型的离散工程优化问题。尤其在基因组数据规模扩大时，搜索空间急剧增长的问题日益显著。本文以基因组的推断为背景，探讨大规模离散工程优化问题的协同算法效果。祖先基因推断是系统进化树重建的关键环节，能够为生物学深层进化模式的发现等很多重要问题提供帮助，构建进化树以寻找不同生物的种间同源基因和种内同源基因，广泛应用于生物学、基因组学和病毒学领域，如对冠状病毒基因组序列进行溯源分析、蛋白质和流行性疾病网络结构预测、药物设计等[12-16]。

基于简约方法进行祖先基因推断被广泛研究。2009 年，Xu[17]提出了一种基于充分子图（AS-Median, adequate subgraph median）的分支定界方法，基于子图的迭代贪心搜索求解祖先基因推断优化问题，但适用于小规模数据集。2015 年，Feijão 等[18]提出一种基于中间基因组的系统发育重建算法，能够在保持性能时花费较低的计算成本。另一方面，计算智能由于启发式信息的搜索特点[19]，也被应用于祖先基因推断的研究中。2013 年，Gao 等[20]提出基于遗传算法的祖先基因推断求解算法（GA-Median），GA-Median 主要集成了遗传算法与基因组分类策略来推断祖先基因。进一步，Gao 等[21]采用协同进化遗传算法，通过分而治之和共享初始节点集合的策略来提升叶子节点规模增大时祖先基因推断的准确性。Xia 等[22]提出基于模拟退火算法的求解方法（SA-Median），SA-Median 的计算成本较低但获取的解性能低于 AS-Median 和 GA-Median。

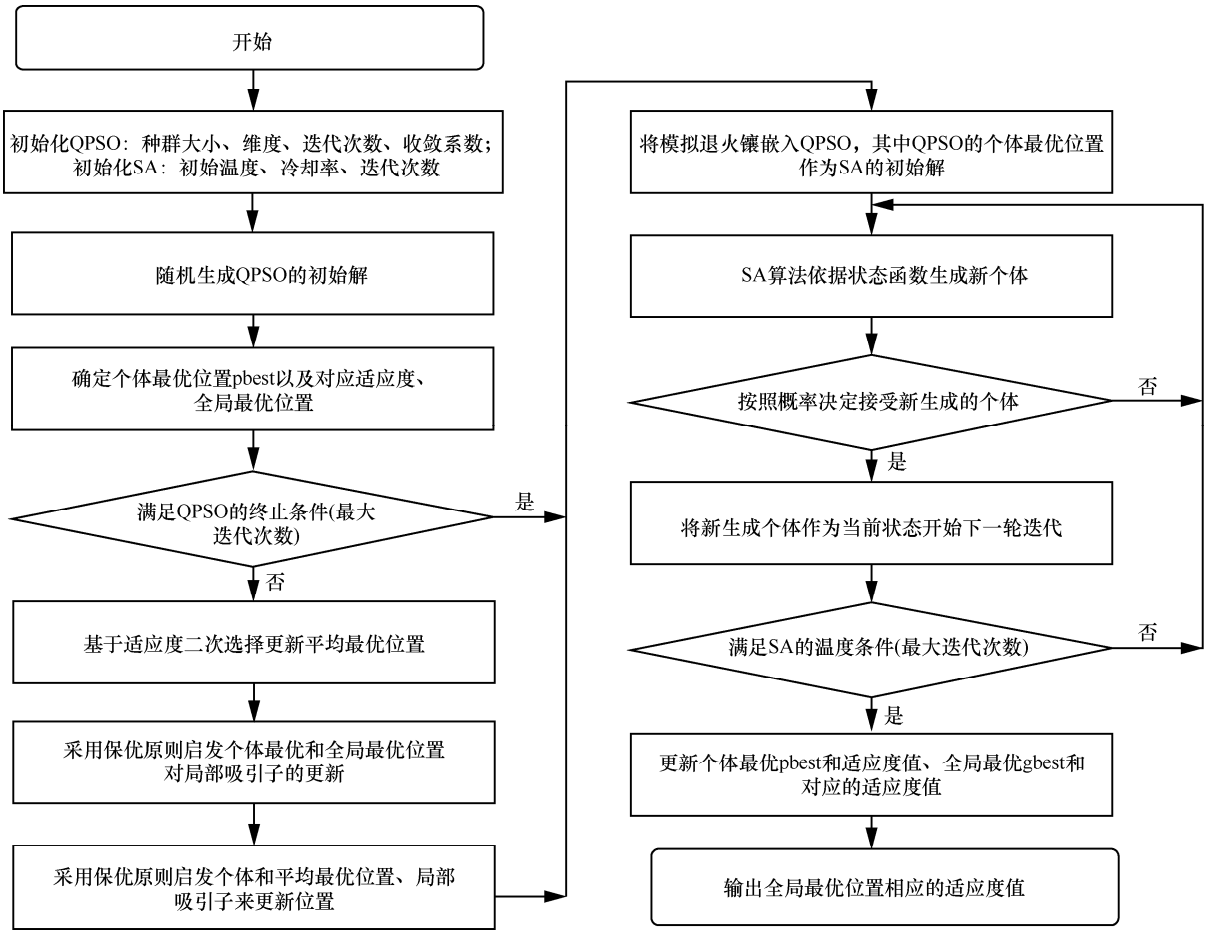

图 1　IDQPSO-SA 算法的流程

面对大规模祖先基因组推断，SA-Median、GA-Median 和 AS-Median 体现出不同特点。SA-Median 具有最低的计算开销，但除时间成本以外的其他性能指标受限。此外，SA-Median 并不具有并行性，进化搜索时由于没有冗余和历史信息从而搜索能力有限。GA-Median 通过生成较多的候选解从而不断进行解的选择来保证所求解质量，但 GA-Median 的计算开销太大。AS-Median 的计算开销和存储空间会随着进化事件的增加而快速增长，对硬件配置的要求随基因组规模和距离的增大而急剧上升。以上分析表明，SA-Median、GA-Median 和 AS-Median 的可扩展性受限，不适用于解决大规模和距离较远的祖先基因推断问题。

针对祖先基因推断应用实例，n 个基因序列包含 $2^n n!$ 个祖先基因组可能性，在大规模、高维、基因组距离较远时整个搜索空间会急剧增长。此时，已有的求解算法在硬件内存、存储空间、计算开销上都面临一定的不足。QPSO 具备较强的全局搜索能力，且 SA 由于突跳性能够在一定程度上避开局部最优。因此，本文提出 IDQPSO-SA，采用基于平均适应度值的二次选择更新全局平均最优位置的策略，克服传统 QPSO 算法无法应用于离散问题的不足，将二次切割与连接（DCJ, double cut joining）排序策略引入 IDQPSO-SA 来降低搜索空间大小、提升祖先基因推断的搜索效率。

3.2　基因进化事件与 DCJ 操作

基因组由有序的带符号的基因序列组成并表示为 $\{g_1, g_2, \cdots, g_i, \cdots, g_j, \cdots, g_n\}$。其中，基因的正向和反向分别由 g 或 $-g$ 表示。对于基因 g_i，头部和尾部分别由 g_i^{h} 和 g_i^{t} 表示。正向表示方向从头到尾 $(g_i^{\mathrm{h}} \to g_i^{\mathrm{t}})$，而反向则表示方向从尾到头 $(g_i^{\mathrm{t}} \to g_i^{\mathrm{h}})$。考虑方向后，基因组可以是线性或圆形（当头和尾重合）形式。

1) 基因组进化事件

基因组进化事件包括倒位（inversion）、转换（transition）、易位（translocation）、裂解（fission）和合并（fission）。采用倒位操作，则该基因组表示

为$\{g_1,g_2,\cdots,g_{i-1},-g_j,-g_{j-1},\cdots,-g_i,g_{j+1},\cdots,g_n\}$。假设$j<k$，并给定3个基因序列$\{g_i,g_j,g_k\}$，当进行转换操作后则生成一个新的基因组$\{g_1,g_2,\cdots,g_{i-1},g_{j+1},\cdots,g_{k-1},g_i,\cdots,g_j,g_k,\cdots,g_n\}$。易位是指当一条染色体的末端断裂时，将其附加到另一条染色体的末端。裂解是指将一条染色体分裂成2条染色体。合并是指将2条染色体合并成一条染色体。

如果g_i紧随着g_j，则定义g_i和g_j相邻，2个连续基因的邻接（adjacency）具有4种类型：$\{g_i^{\mathrm{h}},g_j^{\mathrm{h}}\},\{g_i^{\mathrm{h}},g_j^{\mathrm{t}}\},\{g_i^{\mathrm{t}},g_j^{\mathrm{h}}\},\{g_i^{\mathrm{t}},g_j^{\mathrm{t}}\}$。此外，当2个基因在一个基因组中相邻但在另一个基因组中不相邻，且该端是末端不与任何其他基因相邻时，则产生断点。

2) DCJ距离

DCJ操作由Yancopoulos等[23]提出，包含了所有基因组进化事件。常见的DCJ操作包含以下4种。

① 邻接对$\{g_1,g_2\}$和$\{g_3,g_4\}$可以由邻接$\{g_1,g_3\}$和$\{g_2,g_4\}$或$\{g_1,g_4\}$和$\{g_2,g_3\}$进行重新连接。

② 邻接$\{g_1,g_2\}$和端$\{g_3\}$可以由邻接$\{g_1,g_3\}$和端$\{g_2\}$或邻接$\{g_2,g_3\}$和端$\{g_1\}$进行重新连接。

③ 端$\{g_1\}$和端$\{g_2\}$可以由邻接$\{g_1,g_2\}$进行合并。

④ 邻接$\{g_1,g_2\}$可以裂解成端$\{g_1\}$和$\{g_2\}$。

DCJ距离定义为一个基因组转化为另一个基因组所需进行的DCJ操作数目。不同的DCJ操作会影响奇数边和环的个数，且会进一步影响邻接图的结构，基于邻接和端的关系构建的邻接关系如图2所示。基因组G_1与基因组G_2的进化距离为

$$d_{\mathrm{DCJ}}(G_1,G_2)=n-\left(C+\frac{I}{2}\right) \tag{4}$$

其中，$d_{\mathrm{DCJ}}(G_1,G_2)$表示$G_1$与$G_2$之间的DCJ距离，$n$表示基因组的长度，$C$表示环个数，$I$表示奇数边个数。图2中，给定2个基因组，分别为$G_1=\{g_1,g_2,g_3,g_4,g_5\}$和$G_2=\{g_3,-g_2,-g_1,g_4,g_5\}$。其中，$I=2$，$C=1$，$n=5$。则根据式(4)可得$G_1$和$G_2$的DCJ距离为$n-\left(C+\frac{I}{2}\right)=3$。

3) DCJ Median问题

3个基因组定义了最小的无根二叉树，祖先基因推断问题定义如下：给定3个基因组和一个祖先基因组，若能使该祖先基因组到3个基因组的DCJ距离之和最小化，则该祖先基因组的推断转化为中位基因（Median）问题的求解。因此，祖先基因推断问题称为DCJ Median问题。如图3所示，Median到给定3个基因组的进化距离之和为

$$S_3=d(G_1,G_m)+d(G_2,G_m)+d(G_3,G_m) \tag{5}$$

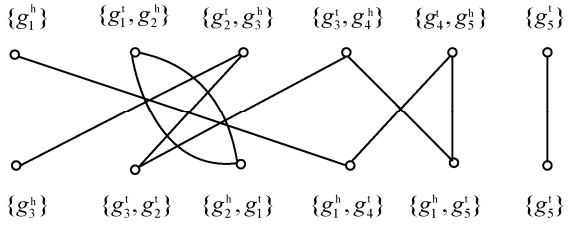

图2 2个基因组的邻接关系

给定3个基因组G_1、G_2和G_3，G_m表示祖先基因组，那么DCJ Median定义为3个给定基因组到祖先基因组的距离之和S_3的最小值。

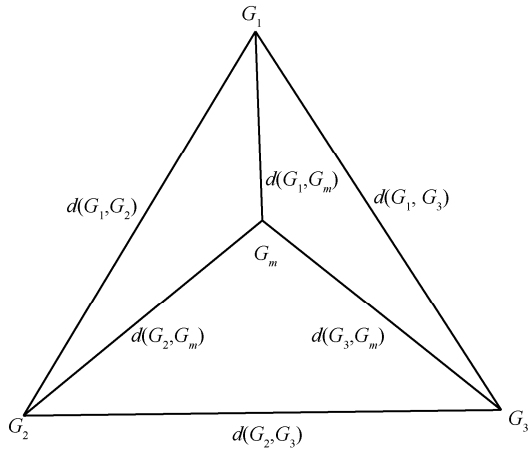

图3 DCJ Median问题

4) DCJ排序策略

由于n个基因序列包含$2^n n!$个祖先基因组可能性，若采取穷尽搜索则计算开销较高，随着数据规模的不断增加，搜索空间快速增长的问题日益显著。因此，若只采用IDQPSO-SA来进行大规模祖先基因离散问题的推断，则会由于计算代价过高而在一段非常长的时间内无法收敛。为了克服计算开销过高的不足，进一步加速搜索进程，本文采用DCJ排序策略引入IDQPSO-SA算法来求解DCJ Median问题（QPSOSA-Median）。

DCJ距离定义为一个基因组转化为另一个基因组所需进行的DCJ操作个数，由于不同的DCJ操作会影响奇数边和环的个数、邻接图的结构，因此不同的DCJ操作的进化成本不一样。祖先基因的推断作为进化重建的关键，其目标是确立一个

Median，其到已给定基因组的进化距离最小。DCJ 排序策略的主要思想如下。针对 2 个基因组，存在多种不同的 DCJ 操作来完成进化，但是不同的 DCJ 操作求得的进化距离是不一样的。从进化操作出发，选择一条能够提升搜索空间效率的方法，若所求解的祖先基因组刚好位于基因组 G_i 转化为 G_j 的路径上，则可以节约搜索空间，从而实现整个进化成本的最小化。该策略被称为 DCJ 排序策略。本文采用的 DCJ 操作都是指最优 DCJ 操作，不包括中性和反最优操作。

3.3　基于 QPSOSA-Median 的祖先基因推断

3.3.1　QPSOSA-Median 的初始化

1) 基于 DCJ 排序的种群初始化

在 QPSOSA-Median 中，整个种群初始化的核心是生成一系列的候选祖先基因组。其中，初始候选解会进一步影响 QPSOSA-Median 的性能。高维时搜索空间较大，若随机选择一个候选祖先基因组则可能造成与真正最优的祖先基因进化距离非常远，从而导致搜索很难收敛。为此，在该 QPSOSA-Median 中引入 DCJ 排序策略。依据从 G_i 到 G_j 的 $\frac{d_{\mathrm{DCJ}}(G_i,G_j)}{10}$, $\frac{2d_{\mathrm{DCJ}}(G_i,G_j)}{10}$, $\frac{3d_{\mathrm{DCJ}}(G_i,G_j)}{10}$, $\frac{4d_{\mathrm{DCJ}}(G_i,G_j)}{10}$, $\frac{5d_{\mathrm{DCJ}}(G_i,G_j)}{10}$, $\frac{6d_{\mathrm{DCJ}}(G_i,G_j)}{10}$ 距离产生 6 个候选 Median，然后随机选择一个作为 QPSOSA-Median 的初始解。

2) 设定进化成本为适应度函数

在 QPSOSA-Median 中，适应度函数反映整个物种进化的好坏，而且物种的适应度值决定是否能够在下一代进化中被保留，其中更优适应度值的物种能够以更大概率在进化过程中生存下来。针对 DCJ Median 问题，一种有效的方法是采用进化距离成本 MS（median score）作为适应度函数，3 个基因组与候选祖先基因组之间的 DCJ 距离成本为

$$F_G = d(G_1,G_m)+d(G_2,G_m)+d(G_3,G_m) \quad (6)$$

其中，F_G 表示给定的 3 个基因组 $\{G_1,G_2,G_3\}$ 的进化距离之和，$d(G_1,G_m)$、$d(G_2,G_m)$、$d(G_3,G_m)$ 分别表示祖先基因组 G_m 到给定基因组 $\{G_1,G_2,G_3\}$ 的进化距离。

3.3.2　基于 DCJ 排序策略的祖先基因组进化更新

1) 基于适应度二次选择的全局平均最优 Median 更新

假设种群规模为 M，并且给定 3 个基因组 $\{G_1,G_2,G_3\}$，候选 Median 基因组总数为 $M\times6\times6$，因此初始候选基因组中有 $36M$ 个候选 Median 基因组，随机选择一个并计算其到给定的 3 个基因组之间的 DCJ 距离。为了提升个体的多样性，QPSOSA-Median 采用的基于适应度二次选择更新平均最优位置的策略是针对整个种群的。首先，对初始候选解的进化距离成本进行平均，保存小于平均进化成本的个体，淘汰大于平均进化成本的个体；然后，对上一轮已经保存个体的进化距离成本第二次求平均；最后，选择与第二次求得的平均进化成本最接近的个体作为全局平均最优祖先基因组。该全局平均最优祖先基因组充分体现了生物基因序列离散化的特点，也反映了整个种群的多样性。

2) 采用 DCJ 排序策略和保优原则更新局部吸引子

QPSOSA-Median 算法中采用 DCJ 排序策略指引和全局最优位置的保优原则更新局部吸引子。更新过程中，参数 μ 设为 0.5，即局部最优 Median 和全局最优 Median 具有相同的权重系数。由于 DCJ 排序策略进行交换和 2 个比较基因组之间的目标顺序有关，目标基因组应选取进化距离成本较低的基因组。由于全局最优在任何情况下都不差于个体局部最优，针对局部最优和全局最优的迭代更新过程，从 pbest 到 gbest 分别以 $\frac{d_{\mathrm{DCJ}}(\mathrm{gbest},\mathrm{pbest})}{10}$, $\frac{2d_{\mathrm{DCJ}}(\mathrm{gbest},\mathrm{pbest})}{10}$, $\frac{3d_{\mathrm{DCJ}}(\mathrm{gbest},\mathrm{pbest})}{10}$, $\frac{4d_{\mathrm{DCJ}}(\mathrm{gbest},\mathrm{pbest})}{10}$, $\frac{5d_{\mathrm{DCJ}}(\mathrm{gbest},\mathrm{pbest})}{10}$, $\frac{6d_{\mathrm{DCJ}}(\mathrm{gbest},\mathrm{pbest})}{10}$ 距离产生 6 个候选基因组，然后选择最优的一个基因组作为候选局部吸引子 P_{id}。该单次 DCJ 排序更新局部吸引子的策略能够确保整个种群都向进化成本最低的方向迭代进化。

3) 采用二次 DCJ 排序策略和保优原则更新 Median

祖先基因组的更新主要包含 2 个阶段，采用个体和平均最优位置、局部吸引子二次保优原则更新个体，并采用 DCJ 排序策略指导最优 Median 基因组的搜索。在第一个阶段，基于 mbest 从左到右将 DCJ 排序策略应用于 X_{id}，其中 X_{id} 表示当前求得的 Median，mbest 表示平均最优候选 Median。在第二个阶段，在 X'_{id} 和 P_{id} 之间应用 DCJ 排序策略来更新

当前 Median。此外，为了增强 QPSO 的搜索能力，QPSOSA-Median 进一步采用 DCJ 排序策略来启发式搜索 Median 基因组，从当前候选 Median 到 3 个给定的基因组之间随机取样，然后从取样中选择一个作为下一代的 Median 基因组。

4) 采用 IDQPSO-SA 算法更新 Median

IDQPSO-SA 算法主要采用镶嵌结构，即将模拟退火算法增加到 QPSO 的更新过程中，种群更新的主要思路为：在初始阶段，当 QPSO 中所有个体实现了进化更新后，将个体最优 Median 作为 SA 的初始解；然后，针对所有个体，依据 SA 的状态产生函数生成新的个体，状态接受函数以一定概率选择接受新个体；SA 进行退火操作，并将更新后的个体传递到下一轮进行迭代；整个 SA 不断迭代直至温度达到设定的终止条件，整个 IDQPSO-SA 算法的混合搜索完成。此时，全局最优 Median 则为整个 IDQPSO-SA 算法推断的祖先基因组。

QPSOSA-Median 中，每个物种分别独立进化，每一次进化促使整个生态系统更好地适应整个环境。随着迭代的不断进化，搜索到的全局最优解越来越接近理论意义上的祖先。当适应度函数评估次数达到终止条件时，整个搜索进程停止，否则重复搜索直至收敛。

4 实验结果与分析

4.1 实验环境和参数设置

QPSOSA-Median 的实验运行环境配置如下：Dell PowerEdge R930 with Xeon E7-4820V4*2 @ 2.10 GHz *24，256 GB 内存和 2 TB 硬盘。数据主要依据不同进化率生成，以 $d=\frac{r}{n}$ 进行变化，其中 d 表示直径，r 表示每个平均分支长度（事件数目），n 表示基因组长度。基因组长度设置为 1 000，平均事件数目变化范围为 550～1 000，相应的进化率变化范围为 0.55～1。针对不同进化率，分别生成 20 个数据集。

所有基于计算的智能算法以相同的适应度评估次数为标准，设为 300 000 次。其中，GA-Median 和 AS-Median 仅运行一轮，因为基于多次实验经验，GA-Median 和 AS-Median 的计算开销较高。虽然种群规模影响 QPSOSA-Median 的性能，但影响有限，且基于实验经验，随着种群规模的提升，计算开销显著增加。因此将本文种群规模设为 20。参数设置如下：QPSO 最大迭代次数和种群规模分别为 1 000 和 20；SA 算法迭代次数为 8 000，初始温度和冷却速率分别为 10 和 0.9；GA-Median 最大迭代次数为 100，每个采样步骤生成 50 个基因组。

4.2 QPSOSA-Median 与 SA-Median 的性能比较

为了评估 QPSOSA-Median 的性能，本文主要考虑以下 4 个性能指标：进化成本、求得的 Median 到真实祖先的进化距离、邻接准确率和运行时间。其中，邻接准确率定义为推断的 Median 基因组和真实祖先之间的交集与其并集的比值，即

$$\mathrm{Acc}\left(G_m,G_t\right)=\frac{G_m\bigcap G_t}{G_m\bigcup G_t} \tag{7}$$

其中，$\mathrm{Acc}\left(G_m,G_t\right)$ 表示邻接准确率，G_m 和 G_t 分别表示推断的 Median 基因组和真实祖先。

QPSOSA-Median 与 SA-Median 的性能比较如表 1 所示。从表 1 可以看出，同 SA-Median 相比，随着进化事件个数的增加，本文提出的 QPSOSA-Median 能够取得较小的进化成本。此外，QPSOSA-Median 减少了与真实祖先之间的进化距离，提升了邻接准确率。然而，由于 QPSOSA-Median 包含多个独立物种分别进行搜索，计算开销比 SA-Median 更加昂贵。此外，同 SA-Median 相比，QPSOSA-Median 在大部分情况下性能都较优，且不随进化事件个数的增加而增加计算开销，因为计算开销主要和算法的收敛速度有关，QPSOSA-Median 集成了 QPSO 和 SA 的混合算法的全局和突跳性的优势更有助于收敛。综上，QPSOSA-Median 同 SA-Median 相比，能够提升求解 Median 的性能。

4.3 QPSOSA-Median 与 GA-Median 和 AS-Median 的性能比较

1) 进化成本

进化成本是所有指标当中最能反映求得的 Median 基因组与真实祖先之间的关系的，进化成本较低表示性能更优。QPSOSA-Median、GA-Median 和 AS-Median 的进化成本比较如表 2 所示。对表 2 结果进行分析可知，QPSOSA-Median 的进化成本最小，AS-Median 次之，GA-Median 最大。进一步地，同 AS-Median 进行对比分析，QPSOSA-Median 和 AS-Median 之间的差距随着进化事件个数的增加而逐渐增大。从以上分析可得，QPSOSA-Median 非常具有竞争力。

表 1　QPSOSA-Median 与 SA-Median 的性能比较

r/个	进化成本		与真实祖先的进化距离		邻接准确率		运行时间/s	
	QPSO-SA-Median	SA-Median	QPSO-SA-Median	SA-Median	QPSO-SA-Median	SA-Median	QPSO-SA-Median	SA-Median
550	1 545.1	1 600.4	510.1	544.0	2.74×10^{-1}	2.46×10^{-1}	8 014	456
600	1 609.2	1 664.0	573.8	602.7	2.26×10^{-1}	2.08×10^{-1}	4 867	454
650	1 662.3	1 718.4	623.0	653.6	1.95×10^{-1}	1.75×10^{-1}	5 744	445
700	1 696.1	1 750.3	670.8	699.4	1.67×10^{-1}	1.49×10^{-1}	5 471	440
750	1 730.5	1 786.3	711.1	730.2	1.43×10^{-1}	1.33×10^{-1}	4 516	435
800	1 759.5	1 811.4	749.5	767.0	1.20×10^{-1}	1.12×10^{-1}	4 390	427
850	1 783.5	1 832.4	782.3	793.9	1.04×10^{-1}	9.76×10^{-2}	4 424	425
900	1 799.3	1 850.0	801.6	817.6	9.34×10^{-2}	8.64×10^{-2}	4 710	422
950	1 813.8	1 866.2	826.0	839.6	8.18×10^{-2}	7.54×10^{-2}	4 689	419
1 000	1 821.9	1 876.2	846.9	857.9	7.24×10^{-2}	6.60×10^{-2}	4 985	417

表 2　QPSOSA-Median、GA-Median 和 AS-Media 的进化成本比较

r/个	QPSOSA-Median	GA-Median	AS-Median
550	1 545.1	1 714.8	1 551.9
600	1 609.2	1 764.8	1 621.8
650	1 662.3	1 810.0	1 680.0
700	1 696.1	1 835.3	1 719.3
750	1 730.5	1 865.8	1 757.6
800	1 759.5	1 890.4	1 790.9
850	1 783.5	1 908.0	1 814.2
900	1 799.3	1 918.3	1 830.2
950	1 813.8	1 929.9	1 848.0
1 000	1 821.9	1 940.0	1 856.2

表 3　3 种算法推断的 Median 基因组与真实祖先之间的平均进化距离比较

r/个	QPSOSA-Median	GA-Median	AS-Median
550	510.1	545.4	541
600	573.8	595.5	615.9
650	623	639.7	678.9
700	670.8	677.5	726.4
750	711.1	720.1	765.4
800	749.5	754.2	802.6
850	782.3	780.1	831.4
900	801.6	803.6	854.4
950	826	825.3	872.7
1 000	846.9	848.4	888

2) 与真实祖先之间的平均进化距离

3 种算法推断的 Median 基因组与真实祖先之间的平均进化距离比较如表 3 所示。从表 3 可以看出，总体来看，QPSOSA-Median 在不同的基因进化事件数时与真实祖先之间的平均进化距离最小，AS-Median 平均进化距离最大。当进化事件个数为 850 和 950 时，虽然 QPSOSA-Median 求得的 Median 基因组与真实祖先之间的平均进化距离比 GA-Median 大，但是结果和 GA-Median 非常接近。此外，在基因组进化事件相对较小时，QPSOSA-Median 所求得的与真实祖先之间的平均进化距离与 GA-Median 差距较大。

3) 邻接准确率

邻接准确率表示的是祖先基因组与真实祖先之间的交集与并集的比值，邻接准确率越大表明性能越好。QPSOSA-Median、GA-Median 和 AS-Median 的邻接准确率比较如表 4 所示。与 GA-Median 和 AS-Median 相比，针对不同基因进化事件个数，QPSOSA-Median 获取的邻接准确率优于 SA-Median 和 GA-Median。此外，在基因进化事件数目小于或等于 650 时，GA-Median 最差；但当事件数大于 650 时，AS-Median 的性能最差。表 4 充分表明 QPSOSA-Median 在邻接准确率这一指标上具有优越的性能。

表 4 QPSOSA-Median、GA-Median 和 AS-Median 的邻接准确率比较

r/个	QPSOSA-Median	GA-Median	AS-Median
550	2.74×10^{-1}	2.09×10^{-1}	2.72×10^{-1}
600	2.26×10^{-1}	1.84×10^{-1}	2.19×10^{-1}
650	1.95×10^{-1}	1.63×10^{-1}	1.74×10^{-1}
700	1.67×10^{-1}	1.47×10^{-1}	1.46×10^{-1}
750	1.43×10^{-1}	1.29×10^{-1}	1.23×10^{-1}
800	1.20×10^{-1}	1.12×10^{-1}	1.01×10^{-1}
850	1.04×10^{-1}	9.92×10^{-2}	8.51×10^{-2}
900	9.34×10^{-2}	8.89×10^{-2}	7.25×10^{-2}
950	8.18×10^{-2}	8.04×10^{-2}	6.37×10^{-2}
1 000	7.24×10^{-2}	7.06×10^{-2}	5.46×10^{-2}

4) 运行时间

QPSOSA-Median、GA-Median 和 AS-Median 的运行时间如表 5 所示。随着基因进化事件数量的增加，AS-Median 的运行时间急剧增加，当基因进化事件为 1 000 时，需要耗费长达 40 h。其次，为了统计分析，本文生成 20 个实例进行验证，计算一个实例需要耗费 10 h 以上，整个实例则需要耗费 9 天。GA-Median 的计算代价太高。综合对比，QPSOSA-Median 计算开销比 GA-Median 和 AS-Median 低得多。

表 5 QPSOSA-Median、GA-Median 和 AS-Median 的运行时间

r/个	QPSOSA-Median/s	GA-Median/s	AS-Median/s
550	8 014	33 032	33 496
600	4 867	32 908	48 625
650	5744	32818	67 486
700	5 471	32 735	96 020
750	4 516	32 676	107 456
800	4 390	32 528	123 077
850	4 424	32 478	131 336
900	4 710	32 445	131 510
950	4 689	32 451	137 438
1 000	4 985	32 420	142 356

5 结束语

为了解决大规模离散工程优化面临的高维等复杂性难题，本文提出一种结合量子粒子群优化的全局搜索和模拟退火的概率突跳性的协同算法。首先，从目标函数求得的适应度值出发，提出一种基于适应度二次选择的全局平均最优位置更新策略，克服传统 QPSO 进化更新方法不适合离散工程优化问题的不足；其次，将 DCJ 排序策略引入 IDQPSO-SA 来降低搜索空间大小。通过在大规模且距离较远的不同基因组进化事件的数据集上的实验表明，本文提出的算法同已有算法相比能够取得较好的性能，这进一步表明了本文算法的有效性和稳定性。

尽管 IDQPSO-SA 算法在不同的进化事件下进行测试验证了性能，但有许多问题值得进一步研究。例如，将本文提出的改进的协同算法用于对大规模优化问题采用的分布式网络进行调度从而减少能耗；对初始解的选取进行优化和采用分布式计算来降低计算开销；结合计算智能算法中的搜索策略来加速搜索过程；采用深度学习来深度挖掘进化历史，从大数据分析的角度为大规模优化问题提供新的见解等；增加协同算法的实际应用，如对冠状病毒基因组序列进行分析以溯源。

参考文献：

[1] 王凌, 沈婧楠, 王圣尧, 等. 协同进化算法研究进展[J]. 控制与决策, 2015, 30(2): 193-202.
WANG L, SHEN J N, WANG S Y, et al. Advances in co-evolutionary algorithms[J]. Control and Decision, 2015, 30(2): 193-202.

[2] 王万良, 张兆娟, 高楠, 等. 基于人工智能技术的大数据分析方法研究进展[J]. 计算机集成制造系统, 2019, 25(3): 529-547.
WANG W L, ZHANG Z J, GAO N, et al. Research progress of big data analytics methods based on artificial intelligence technology[J]. Computer Integrated Manufacturing Systems, 2019, 25(3): 529-547.

[3] GHEYAS I A, SMITH L S. Feature subset selection in large dimensionality domains[J]. Pattern Recognition, 2010, 43(1): 5-13.

[4] 张震, 魏鹏, 李玉峰, 等. 改进粒子群联合禁忌搜索的特征选择算法[J]. 通信学报, 2018, 39(12): 60-68.
ZHANG Z, WEI P, LI Y F, et al. Feature selection algorithm based on improved particle swarm joint taboo search[J]. Journal on Communications, 2018, 39(12): 60-68.

[5] 王晟, 王雪, 毕道伟. 无线传感器网络遗传—禁忌搜索移动代理测量调度方法[J]. 通信学报, 2008, 29(11): 194-199.
WANG S, WANG X, BI D W. Genetic algorithm-tabu search for mobile agents measurement scheduling in wireless sensor networks[J]. Journal on Communications, 2008, 29(11): 194-199.

[6] 叶苗, 王宇平, 代才, 等. 无线传感器网络中新的最小暴露路径问题及其求解算法[J]. 通信学报, 2016, 37(1): 49-60.
YE M, WANG Y P, DAI C, et al. New minimum exposure path problem and its solving algorithm in wireless sensor networks[J]. Journal on Communications, 2016, 37(1): 49-60.

[7] GHEYAS I A, SMITH L S. Feature subset selection in large dimensionality domains[J]. Pattern Recognition, 2010, 43(1): 5-13.

[8] SUN J, FENG B, XU W. Particle swarm optimization with particles having quantum behavior[C]//Proceedings of the 2004 Congress on Evolutionary Computation. Piscataway: IEEE Press, 2004: 325-331.

[9] LI L, JIAO L, ZHAO J, et al. Quantum-behaved discrete multi-objective particle swarm optimization for complex network clustering[J]. Pattern Recognition, 2017, 63: 1-14.

[10] LUKEMIRE J, MANDAL A, WONG W K. d-QPSO: a quantum-behaved particle swarm technique for finding d-optimal designs with discrete and continuous factors and a binary response[J]. Technometrics, 2019, 61(1): 77-87.

[11] KIRKPATRICK S. Optimization by simulated annealing: quantitative studies[J]. Journal of Statistical Physics, 1984, 34(5-6): 975-986.

[12] LU R, ZHAO X, LI J, et al. Genomic characterisation and epidemiology of 2019 novel coronavirus: implications for virus origins and receptor binding[J]. The Lancet, 2020, 395(10224): 565-574.

[13] WU A, PENG Y, HUANG B, et al. Genome composition and divergence of the novel coronavirus (2019-nCoV) originating in China[J]. Cell Host & Microbe, 2020, 27(3): 325-328.

[14] WANG S W, BITBOL A F, WINGREEN N S. Revealing evolutionary constraints on proteins through sequence analysis[J]. PLoS Computational Biology, 2019, 15(4): e1007010.

[15] WANG Y K, BASHASHATI A, ANGLESIO M S, et al. Genomic consequences of aberrant DNA repair mechanisms stratify ovarian cancer histotypes[J]. Nature Genetics, 2017, 49(6): 856.

[16] TOOSI H, MOEINI A, HAJIRASOULIHA I. BAMSE: Bayesian model selection for tumor phylogeny inference among multiple samples[J]. BMC bioinformatics, 2019, 20(11): 282.

[17] XU A W. A fast and exact algorithm for the median of three problem: a graph decomposition approach[J]. Journal of Computational Biology, 2009, 16(10): 1369-1381.

[18] FEIJÃO P. Reconstruction of ancestral gene orders using intermediate genomes[J]. BMC bioinformatics, 2015, 16(14): S3.

[19] 王万良. 人工智能及其应用(第 4 版) [M]. 北京: 高等教育出版社, 2020.
WANG W L. Artificial intelligence: principles and applications[M]. 4rd Ed, Beijing: Higher Education Press, 2020.

[20] GAO N, YANG N, TANG J. Ancestral genome inference using a genetic algorithm approach[J]. PLoS One, 2013, 8(5): e62156.

[21] GAO N, ZHANG Y, FENG B, et al. A cooperative co-evolutionary genetic algorithm for tree scoring and ancestral genome inference[J]. IEEE/ACM Transactions on Computational Biology and Bioinformatics, 2015, 12(6): 1248-1254.

[22] XIA R, LIN Y, ZHOU J, et al. A median solver and phylogenetic inference based on double-cut-and-join sorting [J]. Journal of Computational Biology, 2018, 25(3): 302-312.

[23] YANCOPOULOS S, ATTIE O, FRIEDBERG R. Efficient sorting of genomic permutations by translocation, inversion and block interchange[J]. Bioinformatics, 2005, 21(16): 3340-3346.

[作者简介]

张兆娟（1990− ），女，江西九江人，浙江工业大学博士生，主要研究方向为大数据、分布式优化、深度学习等。

王万良（1957− ），男，江苏高邮人，博士，浙江工业大学教授、博士生导师，主要研究方向为人工智能、大数据分析、优化调度、计算机智能自动化等。

唐继军（1971− ），男，湖南常德人，博士，天津大学特聘教授、博士生导师，主要研究方向为深度学习、生物信息、高性能计算等。