第 10 卷 第 5 期
2021 年 9 月
集 成 技 术
JOURNAL OF INTEGRATION TECHNOLOGY
Vol. 10 No. 5
Sep. 2021

引文格式：
黄祖成, 沈梦圆, 侯至丞, 等. 基于多路径广度优先搜索算法的代谢通路设计与实现 [J]. 集成技术, 2021, 10(5): 72-79.
Huang ZC, Shen MY, Hou ZC, et al. Design and implementation of metabolic pathway based on multi-path breadth-first searching algorithm [J]. Journal of Integration Technology, 2021, 10(5): 72-79.

基于多路径广度优先搜索算法的代谢通路设计与实现

黄祖成[1] 沈梦圆[2] 侯至丞[1*] TOKUYASU Taku Andrew[3*] 蒙海林[2*]

[1]（广州中国科学院先进技术研究所 机器人与智能装备研究中心 广州 511458）
[2]（广州中国科学院先进技术研究所 生物工程研究中心 广州 511458）
[3]（中国科学院深圳先进技术研究院 深圳合成生物学创新研究院 中国科学院定量工程生物学重点实验室 广东省合成基因组学重点实验室 深圳 518055）

摘 要 寻找从底物到产物的可行代谢通路是代谢工程设计中的核心环节。针对复杂代谢网络中代谢通路不唯一的情况以及传统 *K* 条最短路径（*K*-Shortest Path，KSP）算法效率低的问题，该文通过对传统 KSP 算法的优化，引入关键边概念以减少非必要的重复计算；搭建代谢通路设计 Web 平台，使用并行计算方式提升了算法运算性能。最终，通过引入代谢网络图，对改进 KSP 算法的多路径搜索效率进行验证，结果显示较传统 KSP 算法有 5～9 倍的性能提升。

关键词 代谢通路；多路径搜索；合成生物学；算法
中图分类号 TP 399 **文献标志码** A doi: 10.12146/j.issn.2095-3135.20210427005

Design and Implementation of Metabolic Pathway Based on Multi-path Breadth-first Searching Algorithm

HUANG Zucheng[1] SHEN Mengyuan[2] HOU Zhicheng[1*]
TOKUYASU Taku Andrew[3*] MENG Hailin[2*]

[1](*Intelligent Robot & Equipment Research Center*, *Guangzhou Institute of Advanced Technology*, *Chinese Academy of Sciences*, *Guangzhou* 511458, *China*)
[2](*Bioengineering Research Center*, *Guangzhou Institute of Advanced Technology*, *Chinese Academy of Sciences*, *Guangzhou* 511458, *China*)
[3](*Guangdong Provincial Key Laboratory of Synthetic Genomics*, *CAS Key Laboratory of Quantitative Engineering Biology*, *Shenzhen Institute of Synthetic Biology*, *Shenzhen Institute of Advanced Technology*, *Chinese Academy of Sciences*, *Shenzhen* 518055, *China*)
[*]*Corresponding Author*: *zc.hou@giat.ac.cn*; *tokuyasu@siat.ac.cn*; *hl.meng@giat.ac.cn*

收稿日期：2021-04-27 **修回日期**：2021-05-19
基金项目：国家重点研发计划项目（2018YFA0903200, 2018YFA0902900）；国家自然科学基金项目（31870776）；广东省科技计划项目（2019B030316016, 2017A030313149）；广州市科技计划项目（201904010337）；深圳市科技创新委员会项目（KQTD20150331172101 53）
作者简介：黄祖成，硕士，研究方向为多智能体系统；沈梦圆，博士，研究方向为生物信息学；侯至丞（通讯作者），博士，研究方向为多智能体系统，E-mail：zc.hou@giat.ac.cn；TOKUYASU Taku Andrew（通讯作者），博士，研究方向为计算机科学，E-mail：tokuyasu@siat.ac.cn；蒙海林（通讯作者），博士，研究方向为合成生物学，E-mail：hl.meng@giat.ac.cn。

Abstract　To find possible reactions that exist in metabolic networks is essential for metabolic engineering. The *K*-shortest path (KSP) algorithm is a traditional method that is usually used to identify alternative metabolic pathways. To improve the computation efficiency of conventional KSP method, an efficient KSP-based searching method is proposed in this paper. The basic idea is to introduce the critical edge to reduce the redundant calculation. A web-platform is constructed to design metabolic pathways. The parallel computing technique is introduced to improve the computing efficiency. The proposed method is validated on the KEGG metabolic pathways map, and the results show that the proposed method improve the computation efficiency by 5-9 times, compared with the traditional KSP algorithm.

Keywords　metabolic pathway; multi-path searching; synthetic biology; algorithm

Funding　This work is supported by National Key R & D Program of China (2018YFA0903200, 2018YFA0902900), National Natural Science Foundation of China (31870776), Guangdong Science and Technology Project (2019B030316016, 2017A030313149), Guangzhou Science and Technology Project (201904010337), Project of Shenzhen Science Technology and Innovation Commission (KQTD2015033117210153)

1　引　言

代谢通路(Metabolic Pathways)是指生物体内将源代谢物转化为目标代谢物的一系列连续的酶催化反应[1]，如糖酵解通路、三羧酸循环、梭状芽胞杆菌固定二氧化碳的 Wood-Ljungdahl 通路[2]、蓝藻吸收氨的谷氨酰胺合成酶循环(GS-GOGAT)[3]等。在合成生物学领域，常常需要对代谢通路进行设计与分析。代谢通路设计的目标是给定一个或者若干个起始和目标化合物，从代谢数据中找到并返回生物相关的、由酶催化反应构成的、有特定功能和作用的、从起始化合物到目标化合物的可行路径[4]。在过去的十多年里，随着实验技术的进步，高通量组学数据日益增长，越来越多的代谢通路得以解析。以 KEGG[5] 和 MetaCyc[6]为代表的代谢数据库发展迅速，为识别和发现新的合成替代路径带来了新的机遇和挑战。研究人员根据不同物种的代谢数据特点和生物合成的实际应用需求，开发了多个代谢通路设计算法和工具，如 MAPPS[7]、路径搜索系统[8]、novoPathFinder[9]、MRSD[4]、FogLight[10]和 Metabolic Tinker[11]等。这些软件从算法选择上可以归为基于图论[12]、基于化学计量学[13]和基于逆合成[14]等不同类别。利用这些工具，研究人员可方便地从代谢数据中识别、设计出具有潜在价值的代谢通路[15]。

搜索算法是代谢通路设计工具的核心步骤。在代谢网络较大时，现有的代谢设计工具存在搜索时间相对较长的问题。为此，现有的代谢设计工具尝试从数据结构上对算法进行优化[16-17]，或在路径选取策略上进行优化[18]。此外，在多路径搜索算法方面，传统的前 *K* 条最短路径算法(*K*-Shortest Path，KSP)是以 Yen's 算法为代表。该算法是基于 Dijkstra 算法进行演化的版本，在搜索功能上实现了较为有效的前 *K* 条最短路径搜索，但其算法过程从第 3 条路径开始则需要提升。另外，综合来看 KSP 算法的核心思想都是边删除选择法，而过去众多算法围绕数据结构及路径选取策略上进行过优化，也取得了一定的性能提升效果。近年来，由于计算机硬件性能的提升，传统 KSP 算法优势变得不再明显。因此，本文基于 Yen's 算法进行优化改进的 KSP 算法，

在对路径选取策略进行优化的同时，结合多核心 CPU 计算机硬件特点，使用并行计算方式来进一步提升代谢通路搜索算法的运算性能。

2 数据建模

本文主要是针对合成生物学的代谢网络路径搜索进行优化，代谢网络数据来源于 KEGG 通路数据库（https://www.kegg.jp/kegg/pathway.html）。通过 KEGG API 接口批量下载不同物种及参考的代谢反应图的 KGML 文件，并进行解析来提取化合物和反应方向。将代谢通路表示为图，图中的节点表示为代谢通路中的化合物（代谢物），边表示化合物之间的转化（化学反应或代谢反应）。本文设计的代谢网络路径搜索模型针对 KEGG 数据格式的特点进行了专门优化。在不考虑能耗和确保路径正确的前提下，针对可能存在的代谢通路，尽可能搜索多条路径，把代谢网络数据简化以提高搜索的效率。具体处理方法为：把具有通路的化合物之间的 Cost 值设为 1，不连通的化合物之间的 Cost 值设为 max（实验时，考虑到 KEGG 代谢网络参考图中化合物数量约为 3 000 个，设 max＝9999）。所生成的代谢网络图的邻接表如下：

$$G=\begin{bmatrix} 1 & 1 & 9999 & 9999 \\ 9999 & 1 & 1 & 9999 \\ 9999 & 9999 & 1 & 1 \\ 1 & 1 & 9999 & 1 \end{bmatrix}$$

3 算法设计

3.1 传统 KSP 算法

传统 KSP 算法主要以 Yen’s 算法为代表。其中，Yen’s 算法是 Yen 在 1971 年提出并以其名字命名的算法。Yen’s 算法采用了递推法中的偏离路径算法思想，适用于非负权边的有向无环图结构。其主算法步骤如下：

（1）在非负权边的有向无环图 G 中，使用 Dijkstra 算法找出第一条从起点 S 到终点 D 的最短路径 P_1，加入集合 A（已选路径集合）；

（2）从集合 A 中选择最后加入的一条路径 P_n，在 G 中依次删除路径 P_n 中的边（把边的 Cost 值设为最大），并使用 Dijkstra 算法计算出对应的最短路径 P_{Ni} 加入集合 B（候选路径集合）；

（3）从集合 B 中选出一条 Cost 最小的路径 P_m，加入集合 A 并从集合 B 中删除 P_m；

（4）重复步骤（2）～（3）直到集合 A 的数量达到 K，则集合 A 为所求的前 K 条最短路径。

本文将步骤（2）～（3）定义为次短路径搜索（Second Shortest Path Search，SSPS）过程。在 KSP 算法中，最耗时的运算是 Dijkstra 运算，而该算法中的 SSPS 过程存在较多重复的 Dijkstra 计算。因此，在节点数量较多的网络中进行 KSP 运算将会耗费较长时间。本文主要研究如何对 SSPS 过程进行优化以减少不必要的 Dijkstra 运算，从而提高 KSP 算法性能。

3.2 加速选择法

在 Dijkstra 路径搜索算法中，一次只能找出一条最短路径，但 Cost 相同的最短路径可能并不唯一。在 SSPS 过程前，集合 B 中可能已经存在所需要的最短路径。在合成生物代谢网络中，所有连通的边的 Cost 值设为 1，此时 Cost 值相同的路径都认可为是最短路径，但并不设先后顺序。因此，若在集合 B 中存在与集合 A 最后一条路径 Cost 值相同的路径，则可认为是在集合 A 之后的最短路径。此时，可直接从集合 B 中选择出最短路径 P_n，则省去了 SSPS 的计算过程。

如图 1 所示，当前需要搜索第 4 条最短路径时集合 A 中已有 3 条路径，集合 B 中 Cost 值最小的路径与集合 A 的最后一条路径 Cost 值相等，此时直接从集合 B 中选取 Cost 值为 5 的路径加入集合 A 中作为第 4 条最短路径，从而节省

了一轮 SSPS 运算(此例子中可节省 4 次 Dijkstra 运算)。

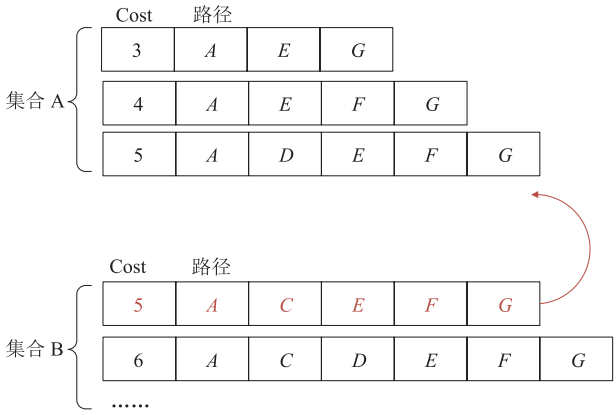

图 1　加速选择法

Fig. 1　Accelerated selection method

3.3　删除重复候选路径

在一个 SSPS 过程中，得出的候选路径可能与上一轮 SSPS 得到的候选路径存在相同的，此时需要判断，当路径已存在集合 B 中则不加入到集合 B。如图 2 所示，删除边 *BC* 或 *CD* 所产生的候选路径是相同的，此时只选择一条加入集合 B 即可。

3.4　记录已选边

在已找到的路径集合 A 中，除第一条路径外，其他路径都有对应删除的边，这些边的集合

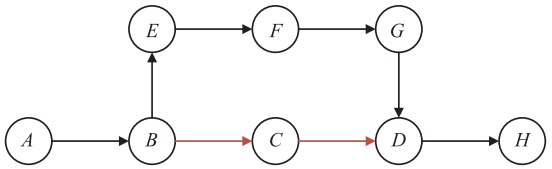

图 2　相同的候选路径

Fig. 2　The same candidate pathways

记为 EA。在进行 SSPS 过程前，先在图 *G* 中删除 EA 里的边，可减少重复的 Dijkstra 运算。

如图 3 所示，集合 A 中第二条最短路径所对应的边为 *BD*，在进行下一轮 SSPS 运算前把边 *BD* 删除以减少 SSPS 的运算时间(此例子中减少了 1 次 Dijkstra 运算)。

3.5　记录关键边

关键边(Critical Edge)是从起点到终点的必经之路，在一个 SSPS 过程中跳过关键边的 Dijkstra 运算可节省运算时间。具体做法为：在 SSPS 过程中，若 Dijkstra 运算的结果 Cost 是最大值，则把对应的边加入到关键边的集合 C 中；随后在下一轮 SSPS 过程中跳过关键边的 Dijkstra 运算，可节省运算时间。

如图 4 所示，当经过一轮 SSPS 运算后，可以得到边 *AB* 和 *DH* 为关键边，并在下一轮 SSPS 运算时把关键边排除，以减少运算量(此例子中

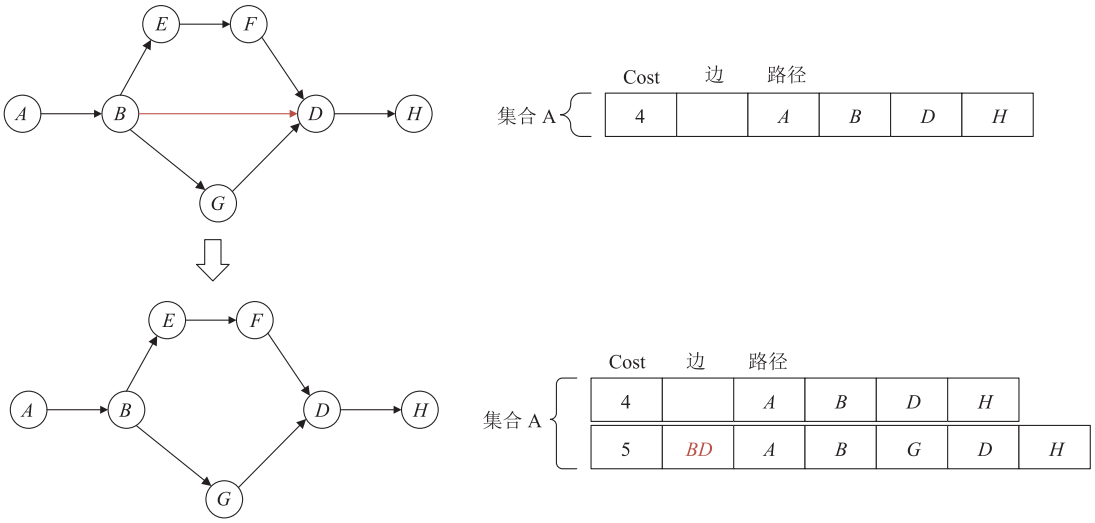

图 3　记录已选边

Fig. 3　Recording the selected edges

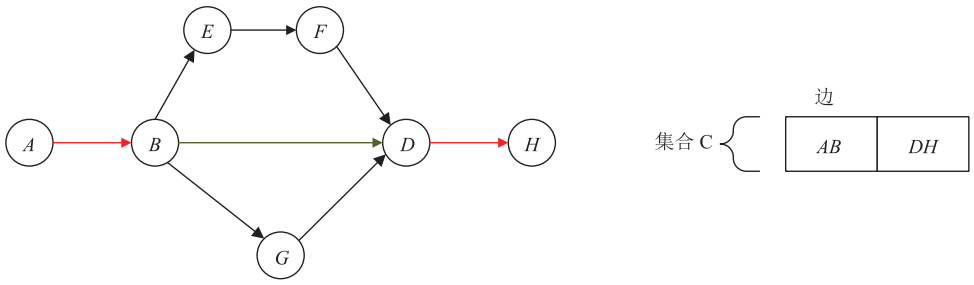

图 4 记录关键边

Fig. 4 Recording the key edges

节省 2 次 Dijkstra 运算)。

4 平台实现

平台采用 Flask＋Vue 框架，实现前后端分离。核心算法代码采用 C++实现以提升运算速率。代谢网络数据主要来源于 KEGG 数据库，在路径搜索前把 KEGG 数据进行初始化(KEGG 数据库转化为邻接表)以便进行 KSP 算法运算。系统主要程序流程如图 5 所示。KEGG 数据的初始化在系统启动前期执行，此部分程序运行时间约为 3～5 s，这对用户来说并不产生影响。对系统用户产生影响的过程为 KSP 运算部分，这也是本文主要的研究工作。

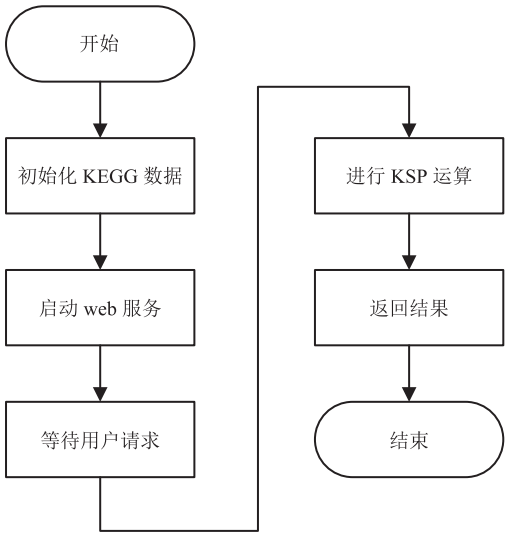

图 5 程序主流程

Fig. 5 The flow chart of the program

图 6 所示为改进的 KSP 算法流程，其中 SSPS 过程采用了并行方式同时进行多个 Dijkstra

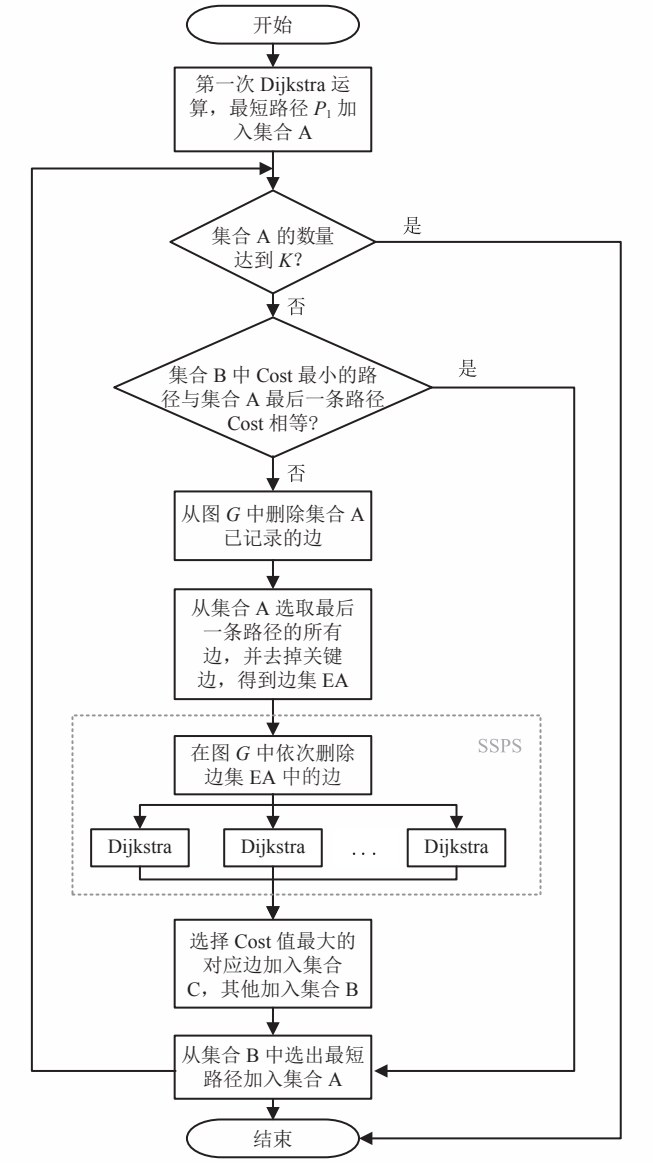

图 6 改进的 KSP 算法流程

Fig. 6 The flow chart of improved KSP algorithm

运算，算法时间复杂度从 $O(n\times m)$ 减少到 $O(n)$，算法结束后会产生 K 条最短路径。为提高用户体验，在实际应用系统中每产生一条最短路径则返回到用户界面上显示，这样可以减少用户主观的等待时间。图 7 所示用户系统界面，搜索得到的路径为按短到长依次显示。

5　算法性能测试

在实验数据中，随机选取起点和终点，分别设定最短路径数量为 1～7 条，观察算法所调用 Dijkstra 的次数来分析算法的性能。具体数据如表 1 所示。

从表 1 可知，当路径数量 K 大于 2 时，改进的 KSP 算法在性能上有 2～3 倍的提升，并行的 KSP 算法在性能上有 5～9 倍的提升，并随着路径数量的增加而不断提升。本次改进的 KSP 算法性能提升较为显著。

表 2 数据为随机选取 4 条目标路径进行前 7 条最短路径的搜索，每条路径的平均搜索时间为 0.27～0.39 s。其中，所选取的 KEGG 参考代谢网络图节点总数为 2 828，边总数为 4 051。即使在 KSP 算法运算时间上增加 0.5 s 的 http 请求及获取化合物及反应信息所需要的时间，在用户端

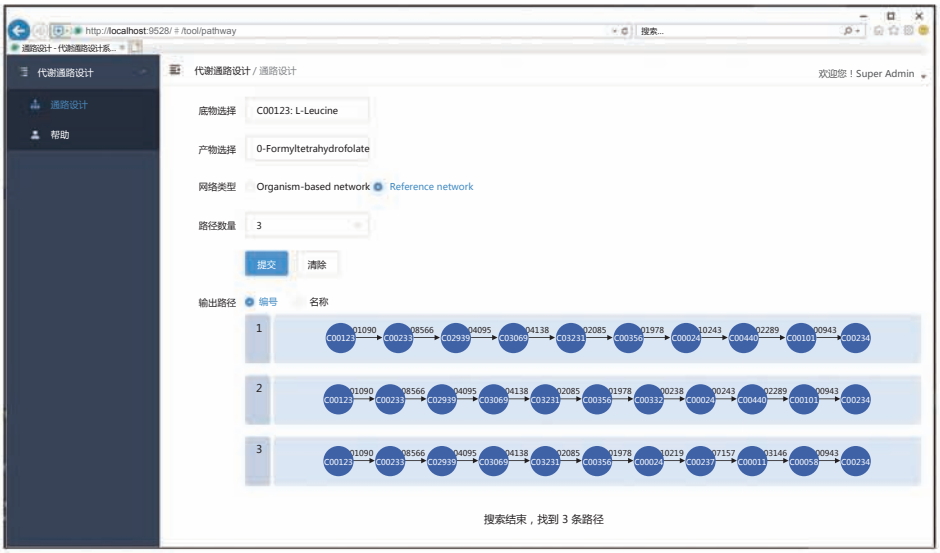

图 7　系统界面

Fig. 7　The system interface

表 1　改进的 KSP 算法性能对比

Table 1　The performance of improved KSP algorithm

路径数量 K	算法性能（Dijkstra 运算次数）			性能提升（倍数）	
	Ken 算法	改进的 KSP 算法	并行的 KSP 算法	改进的 KSP 算法	并行的 KSP 算法
1	1	1	1	1.0	1.0
2	10	10	2	1.0	5.0
3	20	10	3	2.0	6.7
4	30	14	4	2.1	7.5
5	42	14	5	3.0	8.4
6	54	21	6	2.5	9.0
7	66	27	7	2.4	9.4

表 2 改进的 KSP 算法性能实测数据

Table 2 The performance data of improved KSP algorithm

第 K 条路径	运算时间(s)			
	目标路径 1 (C00123→C00234)	目标路径 2 (C00123→C02341)	目标路径 3 (C00233→C02341)	目标路径 4 (C00234→C00544)
1	0.05	0.03	0.03	0.04
2	0.78	0.62	0.64	0.90
3	0	0	0	0.89
4	0.37	0.20	0.28	0.92
5	0	0.28	0.34	0
6	0.61	0.37	0.50	0
7	0.72	0.36	0.42	0
平均	0.36	0.27	0.31	0.39

注：目标路径后面的括号内容(如 C00123→C00234)为 KEGG 数据库中的化合物编号

平均每条路径的搜索总时间仍然小于 1 s，提升了系统的用户体验。

6 讨论与分析

在合成生物学设计尤其是细胞工厂/底盘细胞设计中，代谢通路设计工具可有效辅助研究人员快速找到出发底物及目标产物之间的连接路径，提高设计效率。例如，Yim 等[19]对 1,4-丁二醇(Butane-1,4-diol，BDO)的合成代谢通路进行 1 万多次的计算调查后，设计获得最佳途径，并在大肠杆菌中获得高达 18 g/L 的 BDO。随着下游酶的改善，BDO 滴度提高到 110 g/L。该案例成功展示了代谢通路设计算法在生物合成中的巨大应用潜力。

本文主要针对基于图论的路径搜索算法进行性能优化，用于代谢通路的搜索和发掘。该类算法通过代谢网络中的化合物和化学反应之间的连接关系来搜索可行的代谢通路。目前，该类算法的代表软件主要有 MRSD[4]、FogLight[10]和 Metabolic Tinker[11]等。其主要优点是可以在单一物种或者跨物种中寻找可行的代谢通路，不受物种和反应流平衡等约束；缺点是在找到的代谢通路中容易出现一些连通度高的簇代谢物，而这些簇代谢物的存在会影响整体路径的生物化学意义[11]。

在图论中，路径搜索算法主要有 Dijkstra 算法、A*算法、Bellman-Ford 算法、Floyd-Warshall 算法和 Johnson 算法等。其中，A*算法在有估价函数的条件下可以快速搜索目标路径；Bellman-Ford 算法可以处理含负边值的路径；Floyd-Warshall 算法实现代码简单；Johnson 算法在 Bellman-Ford 算法的基础上优化并提高了稀疏图的运算效率；结合生物学代谢网络中为搜索单源非负边的路径，Dijkstra 算法在时间复杂度上更有优势。自 Dijkstra 算法提出以来，许多学者都对它进行过不同程度的优化以提高其性能。在后来的多路径搜索(KSP)算法上，大部分研究围绕删除路径核心思想进行了许多的改进[16-18]。这些改进的算法在过去计算机性能有限的条件下取得了较好的效果，算法时间复杂度从 $O(n\times n)\rightarrow O(n\times m)\rightarrow O(n\times \log m)$ 不断提升。但随着计算机性能的提升，特别是多核心 CPU 以及多 CPU 架构的计算机系统的广泛应用，传统以单线程进行算法迭代运算的方式并不能发挥很好的效果。本文正是利用了多核心 CPU 的硬件特点，在算法优化的同时采用并行编程方式，把算法的时间复杂度提升到 $O(n)$ 水平，大幅提升了运算性能。

7 结 论

本文针对合成生物学代谢网络中代谢通路非唯一以及传统 KSP 算法效率偏低的问题，对 KSP 算法进行改进优化以提升运算性能。在对 KSP 算法策略上优化的同时，采用并行计算方式进一步提升算法的性能。使用 Python 实现代谢通路设计 Web 平台，并采用 C++编写核心算法的代码。通过引入 KEGG 代谢网络图，验证改进的 KSP 算法比传统 KSP 算法有较大的性能提升，在合成生物学代谢通路设计上具有一定的应用价值。然而，由于代谢反应在不同物种的代谢网络中会有差别，本文基于 KEGG 参考图上进行了搜索算法的研究，并未对物种的特性进行区分。因此，后续工作可在 KEGG 参考图基础上针对不同物种的约束条件进行路径算法研究，以适应合成生物学对不同物种代谢网络通路设计的特定需求。

参 考 文 献

[1] Jeong H, Tombor B, Albert R, et al. The large-scale organization of metabolic networks [J]. Nature, 2000, 407(6804): 651-654.

[2] Wood HG. Life with CO or CO_2 and H_2 as a source of carbon and energy [J]. The FASEB Journal, 1991, 5(2): 156-163.

[3] Chávez S, Lucena JM, Reyes JC, et al. The presence of glutamate dehydrogenase is a selective advantage for the Cyanobacterium *Synechocystis* sp. strain PCC 6803 under nonexponential growth conditions [J]. Journal of Bacteriology, 1999, 181(3): 808-813.

[4] Xia DG, Zheng HR, Liu ZQ, et al. MRSD: a web server for metabolic route search and design [J]. Bioinformatics, 2011, 27(11): 1581-1582.

[5] Minoru K, Miho F, Mao T, et al. KEGG: new perspectives on genomes, pathways, diseases and drugs [J]. Nucleic Acids Research, 2017, 45(D1): D353-D361.

[6] Caspi R, Altman T, Billington R, et al. The MetaCyc database of metabolic pathways and enzymes and the BioCyc collection of pathway/genome databases [J]. Nucleic Acids Research, 2014, 42(D1): D459-D471.

[7] Rizwan RM, Preston GM, Mithani A. MAPPS: a web-based tool for metabolic pathway prediction and network analysis in the postgenomic era [J]. ACS Synthetic Biology, 2020, 9(5): 1069-1082.

[8] Otero-Muras I, Carbonell P. Automated engineering of synthetic metabolic pathways for efficient biomanufacturing [J]. Metabolic Engineering, 2021, 63: 61-80.

[9] Ding SZ, Tian Y, Cai PL, et al. novoPathFinder: a webserver of designing novel-pathway with integrating GEM-model [J]. Nucleic Acids Research, 2020, 48(W1): W477-W487.

[10] Khosraviani M, SahebZamani M, Bidkhori G. FogLight: an efficient matrix-based approach to construct metabolic pathways by search space reduction [J]. Bioinformatics, 2016, 32(3): 398-408.

[11] McClymont K, Soyer OS. Metabolic tinker: an online tool for guiding the design of synthetic metabolic pathways [J]. Nucleic Acids Research, 2013, 41(11): 113.

[12] Pitkänen E, Jouhten P, Rousu J. Inferring branching pathways in genome-scale metabolic networks [J]. BMC Systems Biology, 2009, 3(1): 103.

[13] Chowdhury A, Maranas CD. Designing overall stoichiometric conversions and intervening metabolic reactions [J]. Scientific Reports, 2015, 5(1): 16009.

[14] Hadadi N, Hatzimanikatis V. Design of computational retrobiosynthesis tools for the design of *de novo synthetic* pathways [J]. Current Opinion in Chemical Biology, 2015, 28: 99-104.

[15] Wang L, Dash S, Ng CY, et al. A review of computational tools for design and reconstruction of metabolic pathways [J]. Synthetic and Systems Biotechnology, 2017, 2(4): 243-252.

[16] Azevedo JA, Santos Costa MEO, Madeira J, et al. An algorithm for the ranking of shortest paths [J]. European Journal of Operational Research, 1993, 69(1): 97-106.

[17] De Queirós Vieira Martins E, dos Santos JLE. A new shortest paths ranking algorithm [J]. Investigacao Operacional, 2000, 20(1): 47-62.

[18] Jiménez VM, Marzal A. Computing the *K* shortest paths: a new algorithm and an experimental comparison [M] // Algorithm Engineering. WAE 1999. Lecture Notes in Computer Science, 1999, 1668: 15-29.

[19] Yim H, Haselbeck R, Niu W, et al. Metabolic engineering of *Escherichia coli* for direct production of 1,4-butanediol [J]. Nature Chemical Biology, 2011, 7(7): 445-452.