

参数计算中使用的若干技术

周 星 彭 伟

(国防科学技术大学计算机学院 长沙 410073)

摘 要 参数计算和复杂性是上世纪末本世纪初兴起的一门技术。该技术因具有坚实的理论基础和良好的运行效果,故仅经过短短十几年的发展,已经取得了大量的成就。从趋势上看,参数计算已经成为理论计算机科学中越来越热门的一个分支。先简要介绍参数计算和复杂性的重要理论基础和主要思想;之后重点介绍参数计算中使用的主要技术,分析每一种技术的功能作用、基本设计原则和优缺点,并结合简单案例加以说明。

关键词 参数计算,复杂性, NP 完全,核心化,分支搜索

中图法分类号 TP301.6 文献标识码 A

Several Techniques Used in Parameterized Computation

ZHOU Xing PENG Wei

(School of Computer, National University of Defense Technology, Changsha 410073, China)

Abstract Parameterized computation and complexity are one of the techniques arised in the end of last century and the beginning of the current one. Because of its solid theoretical foundation and excellent running performance, it has achieved much success despite of its short-time development. In a trend, parameterized computation is becoming a hotter and hotter branch of computer science. This paper first introduces the important theoretical base and key idea of parameterized computation, then mainly introduced the key techniques used in it, introduced its function, basic design principles and advantages/disadvantages for every technique, and used simple examples to explain it.

Keywords Parameterized computation, Complexity, NP-complete, Kernelization, Branch and search

1 引言

NP 完全理论^[1]是理论计算机科学领域取得的一项重要成果,它为一些难解的问题的研究提供了理论基础,划分了计算难解的界限。它将原理上可计算的问题,划分为 P 类问题和 NP 类问题。在 NP 完全理论提出后的 40 多年时间里,它一直占据着算法分析和设计者的研究的中心位置。至第一个 NP 完全问题被证明之后,到目前为止已经宣布了几千个 NP 完全问题。一旦某个问题被证明是 NP 难的,则可提醒人们该问题是无法用现代计算机有效求解的,从而避免了人们为此问题付出的大量无谓的努力。

过去解决 NP 难问题的方法,包括了贪心算法、近似算法^[3,4]、启发式算法^[5,6]和随机算法^[7,8]。但这几类算法都有其局限性,例如近似算法无法保证得到最优解,只能得到最优解附近范围之内的解,而有的问题获得近似解毫无意义,比如 SAT 问题,启发式算法虽然在实现中往往能得到一些较优秀的解,但是无法保证不会得到很差的解,而且启发式算法的设计没有统一的框架,需要算法设计者具有深厚的经验和睿智深刻的眼光,随机算法则无法保证在确定的时间内给出确信的答案,而且随机算法的运行效率、运行效果,很大程度上依赖于问题的解的分布和随机算法本身,具有很大的不确定性。

近年来,参数计算和复杂性理论^[2,9]得到了很大的发展,该理论来源于这样一个经验:很多 NP 难解问题存在一个或多个参数在较小的范围内变化,或者对无参数的 NP 难解问题可以通过引入一个或多个小范围参数来解决。充分利用小参数这一数学特征,使算法的复杂度尽量依赖于这些小参数,则许多理论上难解的问题可以在实践上有效地解决,这就是参数计算的基本思想。参数计算大部分用来解决判定问题,例如,对于一个顶点覆盖问题(vertex cover, 选定最少的顶点,使得图中的每条边的两个顶点至少有一个被选定),可以引入参数 k , 表示 k 个顶点能否覆盖所有的边,则顶点覆盖问题可以通过判定问题来解决。在某些现实问题中, k 往往范围不大。一个例子是 ETH 的达尔文项目组^[10,11]中,所遇到的一个基因冲突问题可以转化为顶点覆盖问题,因为问题的生物特性,点覆盖的规模 k 不超过 60, 该项目组使用点覆盖问题的目前最优算法^[12]——复杂度是 $O(1.285^k + kn)$ 的参数计算算法(其中 n 是图中的顶点个数),将问题在极短的运行时间内得到正确解决。

参数计算方法已经成功地应用到人工智能^[13]、生物信息计算^[14]、社会学^[15,16]、数据库^[17]、密码学^[18]等领域,其专门国际会议从 2000 年起已成功举办多届,有关这方面的研究论文也大量发表在各种重要的期刊和会议专集上。国内的中南大

本文受国家自然科学基金项目(1070199, 61202486, 61272010)资助。

周 星(1990—),男,硕士生,主要研究方向为算法设计与分析, E-mail: usb8023@gmail.com; 彭 伟 男,博士,研究员,主要研究方向为算法、网络拓扑测量、网络服务质量等。

学、复旦大学、上海交通大学分别设置了专门的研究小组,相关项目研究受到了 973 和自然科学基金的支持。

2 参数计算和复杂性理论的基础

定义 1(参数化问题)^[2] 参数化问题 Q 是一个判定型的问题,是 $\Sigma^* \times N$ 的一个子集,其中, Σ 是一个固定的字符集, N 是所有非负整数的集合。 Q 中的每一个实例都可以用一个形如 (x, k) 的二元组来表示,其中, x 是一个具体的问题, k 是一个非负整数,通常被称为参数。

对于一个算法 A ,如果输入问题 Q 的任意一个实例 (x, k) , A 都能够正确判定实例 (x, k) 是否为 Q 的一个真实例,那么就称 A 为求解参数化问题 Q 的一个算法。进一步地,如果算法 A 的计算复杂性是由输入规模 $|x|$ 和参数 k 的值来度量,那么就称算法 A 是问题 Q 的一个参数化算法^[2]。

定义 2(固定参数可解)^[2] 如果一个参数化问题 Q 能够被一个运行时间为 $f(k)|x|^c$ 的算法 A 求解,其中, f 是一个递归函数, c 是一个不依赖于参数 k 和输入规模 $|x|$ 的常量,那么就称参数化问题 Q 是固定参数可解的(fixed-parameter tractable),算法 A 被称为问题 Q 的一个固定参数可解算法。通常用 FPT 来表示所有固定参数可解问题的集合。

固定参数可解(或者确定参数可解)问题,代表了一类求解算法与参数 k 具有特殊关系的问题。独立集问题(给定一个图 G 和整数 k ,问 G 是否存在 k 个顶点,这些顶点互不相邻)的当前最好的算法^[19]复杂度是 $O(n^{0.792k})$,其不是固定参数可解的,因为这里的指数与 k 相关,事实上独立集问题属于固定参数不可解(fixed parameter intractable)中的 $W[1]$ 类。FPT 并不是对 NP 完全性的重新分类,而是从另一个视角,即从实践的角度来看待问题的难易,两者关系见图 1。

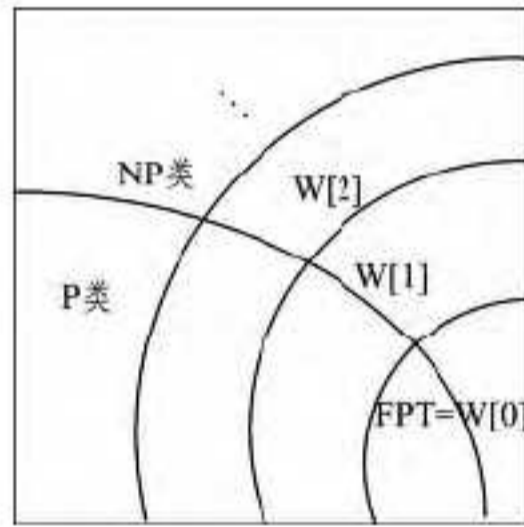


图 1 P 类、NP 类、FPT 和 W 框架的关系

我们在图中认为 $P \subset NP$ (也是当前大家普遍的看法)。基于 FPT 和 W 框架,可以对各领域的难解问题的难解性进行更细致的划分,其中点覆盖问题、顶点反馈集问题、边支配集问题^[20]、d-Hitting Set^[21,22] 问题、Multicut 问题^[23] 都是固定参数可解的。

定义 3(可核心化, kernelization)^[2] 如果存在一个多项式时间的算法 K 和一个递归函数 g ,使得对于参数化问题 Q 的任意一个实例 (x, k) ,算法 K 都能产生 Q 的另一个实例 (x', k') ,其中 $x' \leq g(k)$ 且 $k' \leq k$,并满足 (x, k) 是参数化问题 Q 的一个真实例当且仅当 (x', k') 也是 Q 的一个真实例,那么就称参数化问题 Q 是可核心化的,并且称算法 K 为核心化算法,称实例 (x', k') 为参数化问题 Q 的一个核。

核心化是一种使得规模较大问题变小的预处理技术^[1],核心化后,规模是参数 k 的一个函数,而且新的参数 k' 不比 k 大。核心化后,至少可用枚举法求解问题,复杂度是 k 的指数

形式,即可核心化,则固定参数可解。反过来也是如此^[2]。

由上可知,当一个问题被证明是固定参数可解的,那么一般可以找到它的一个核。将问题缩减之后,再利用包括分支搜索技术在内的各种技术来求解问题。下面的内容将对几种主要的技术进行阐述说明。

3 固定参数可解的主要技术

有关 FPT 算法设计技术的研究一直都是参数计算理论中的一个研究热点。目前主要用到的技术有核心化技术^[26](Kernelization)、分支搜索技术^[25]、动态规划技术等,如图 2 所示,大致反映了算法的设计层次。

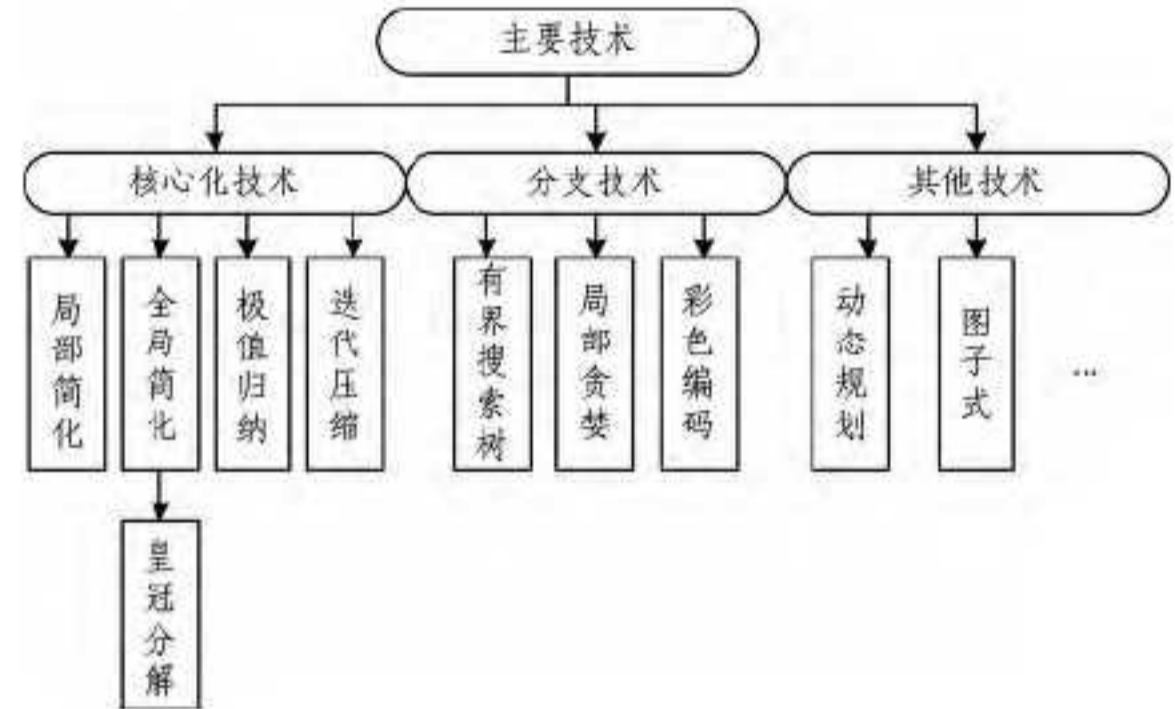


图 2 参数计算中的主要技术

本文第 1 节主要介绍核心化技术的方法和例子;第 2 节主要介绍分支搜索技术;第 3 节主要介绍其他几种非常重要但是比较杂的技术。

3.1 核心化技术

核心化过程是很多参数化问题求解的第一步,也是很多问题求解的“必须”阶段^[27,28]。它的基本思想是先处理问题的一些容易的部分^[29],使得问题缩小,再处理真正难解的部分,即所谓的核。核心化方法包括局部简化、全局简化、极值归纳和随机核心化方法。

3.1.1 局部简化(local reduction)

局部简化技术是一系列可在多项式时间内完成的简化规则,通过应用这些规则能将问题简化到具有固定大小,这样的规则称为局部简化规则^[30]。

以参数化顶点覆盖问题(k -VC)为例,说明局部简化技术的使用:(1)度数为 0 的顶点可以直接删除。(2)度数为 1 的点,把它的邻接点算入覆盖集,而将该点和与其相邻的边直接删除。(3)度数为 2 的点,如果与其相邻的两个点之间有一条边,则将这两个相邻点算入覆盖集,删除这 3 个点和与其相邻的边;如果与其相邻的两个点之间没有边,则可以证明,把该点和两条相邻的边删除,把两个邻接点合并成一个新的点,如果原图有一个大小为 k 的覆盖集,当且仅当经过这个“折叠”操作之后得到的新图拥有一个大小为 $k-1$ 的顶点覆盖集。(4)度数大于 $k+1$ 的点,显然要算入顶点覆盖集,否则覆盖集的大小将超过 k ,将这样的点和邻接的边删除后, k -VC 问题的参数减小到 $k'=k-1$ 。

经过以上几个局部简化之后,新实例的规模不超过 $O(k'^2)$ ^[32],这里的 k' 是新实例的参数,显然 $k' \leq k$,对比定义 3 可知,我们对问题进行了核心化操作。陈等人在文献^[12]中还考虑了顶点的度为 3、4 的局部简化规则。为了避免枚举多种繁琐情况,陈^[31]等人挖掘了这些情况之间的内在联系,将

多种情况统一起来,便于编程实现。

3.1.2 全局简化(global reduction)

全局简化不是在本地图中寻找固定大小的某种结构将其化简,它是将整个问题进行处理,得到有关全局的信息之后将其化简。全局简化的代表是 NT 算法^[33]和皇冠分解^[34],Chelbfc 等人^[35]证明了 NT 算法其实是一个一次性的皇冠分解。一个皇冠结构如图 3 所示。

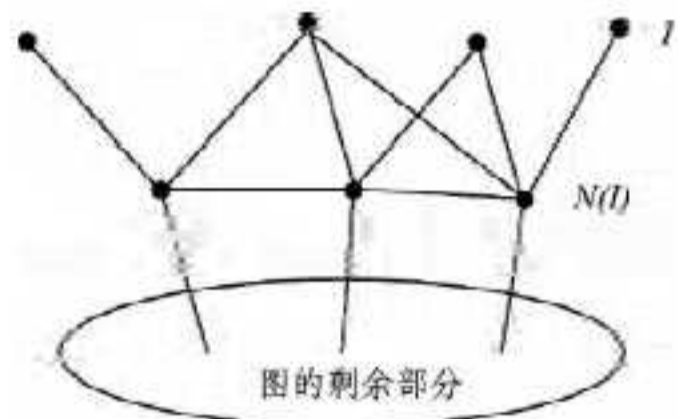


图 3 一个皇冠结构

皇冠结构的顶点由图中的一个独立集 I (图 3 中最上面的顶点)、 I 的邻居节点 $N(I)$ 和图中的剩余部分顶点组成。 $N(I)$ 将 I 和剩余部分分割开,且 $N(I)$ 是 $I \cup N(I)$ 的导出子图的顶点覆盖。

引理 1^[36] 如果图 G 的一个独立集为 I ,且 I 为邻居节点 $N(I)$,若 $|N(I)| \leq |I|$,则 G 中存在一个皇冠。

Faisal 的皇冠分解算法^[32]首先在图中寻找一个极大匹配 M ,未匹配到的点集就是图的一个独立集,然后将此独立集和其邻居按某种方式增长,最后得到一个极大的皇冠。把 $N(I)$ 纳入顶点覆盖集里,删除 I 和 $N(I)$ 称为删除皇冠。重复删除极大皇冠,最后得到核。文献^[32]证明了,皇冠操作后得到的核的大小最多为 $3k$,而寻找皇冠并简化的操作复杂度为 $O(n^{2.5})$ 。

皇冠分解在参数计算中获得了广泛使用,且皇冠分解的核心化效果比局部简化技术更加明显。皇冠分解其实是解决整数线性规划问题时松弛得到实数解,通过实数解构造近似整数解的数学过程的图论表示,但是皇冠分解的实现有时比线性规划慢^[26]。

3.1.3 极值归纳(extremal induction)

设 Q 是一个最大值参数化问题,且其关于参数单调,即若 (x, k) 是一个 no 实例,则 $(x, k+1)$ 也是一个 no 实例。极值归纳的主要思想是先假设已经简化后的实例 (x, k) 是一个 yes 实例,而 $(x, k+1)$ 是一个 no 实例,则 x 会满足一个与 k 的关系,常常是 $|x| \leq g(k)$, g 称为一个边界函数。如果 (x, k) 是一个 yes 实例,则 $|x| \leq g(k)$,这就是界值引理,且 (x, k) 实例的核可以根据推导 $g(k)$ 的过程构造出来。下面以顶点不相交的 $K_{1,3}$ -packing 为例($K_{1,3}$ 是一个 star,即一个中心点与 3 个邻接点通过边相连)加以说明。

规则 1 如果两个相邻顶点 u, v ,其度都不超过 2,则 u, v 这条边可以删除,因为这条边不可能是某个 $K_{1,3}$ 的边。该规则是显然的。

规则 2 如果图中存在一个顶点 v ,其度数 $> 4k-1$,则 (G, k) 是一个 yes 实例当且仅当 $(G[V \setminus v], k-1)$ 是一个 yes 实例。

充分性,如果 v 没有在 k 个 star 中的任何一个,那么 $(G[V \setminus v], k)$ 是 yes 实例,自然 $(G[V \setminus v], k-1)$ 是一个 yes 实例;如果 v 在某个 star 中, v 不能被用两次,那么 $G(V \setminus v, E)$ 中肯定还有 $k-1$ 个 star。

必要性,如果 $G(V \setminus v, E)$ 有 $k-1$ 个 star,则总顶点个数为 $4 * (k-1)$,因为点 v 的度大于 $4 * k-1$,至少还剩余顶点 $(4 * k-1) - 4 * (k-1) = 3$,现在添加了 v 及其度,所以 v 可以和剩余的 3 个顶点组成一个 $K_{1,3}$ star。

Sloper 等人在文献^[26]中证明了,这样得到的核的大小为 $48 * k^2 - 8k$,这就是界函数 $g(k)$ 。极值归纳技术在许多问题上得到了成功的应用,如:Cluster Editing ($4k$ 核)、 k -Internal Spanning Tree ($O(k^3)$ 核)以及 $K_{1,3}$ packing ($O(k^2)$ 核)等。极值归纳技术的关键在于导出界值引理。

3.1.4 迭代压缩(iterative compression)

迭代压缩技术由 Reed 等人^[38]最先提出,事实证明迭代压缩技术是解决单调最小化 NP 难优化这类问题的非常有效的技术。其原理是递归地解决实例,例如 $(G-v, k)$,因为它是判定问题,如果该实例的回答是 no,则 (G, k) 的回答也是 no。但如果 $(G-v, k)$ 是一个真实例,我们能利用它的解来构造求解 (G, k) 的解或者判断 (G, k) 是假实例。因为算法递归调用自身,而调用的次数最多为 $|V(G)|$ 次,所以复杂度为 $|V(G)|$ 乘以处理每一次递归的复杂度。以 k -反馈顶点集问题(是否存在大小不超过 k 的顶点集合 F ,使得原图顶点删除 F 后图中无圈,即图中的每个圈至少有一个顶点在 F 中)为例,简要说明思想:

首先求得 G 的一个导出子图 G_0 的 k -反馈顶点集 F 。现假设往 G_0 里面添加了一个点 v_0 ,显然 $F = F \cup v_0$ 是新图的一个反馈顶点集,从这 $k+1$ 大小的顶点集中构造 k 大小的反馈顶点即可。具体构造方法可参考文献^[39]。文献^[40,41]使用的是更朴素的算法。

核心化的过程就是一组化简规约规则应用的过程,因此化简规约规则的设计是核心化技术的关键。规约规则有的比较明朗,但是有的比较隐蔽,需要对所研究的问题的结构、属性进行细致分析,利用推理、归纳等方法手段才能得出。

3.2 分支搜索算法(Branch and search algorithms)

分支搜索算法是把原实例变成若干个不相交的新实例,任何一个新实例的 yes 回答都能导致原实例的 yes 回答。分支算法包括有界搜索树、彩色编码、局部贪婪法等。

3.2.1 有界搜索树(Bounded search tree)

有界搜索树的模型如图 4 所示,搜索树的每一个节点代表了一个实例,树的每一条边表示父节点实例到子节点实例的转化,子节点与父节点的元素和参数会有所不同。在任意一个父节点处,根据该实例内的某一个局部结构,每一种分支导出了一个与其相对应的新实例。

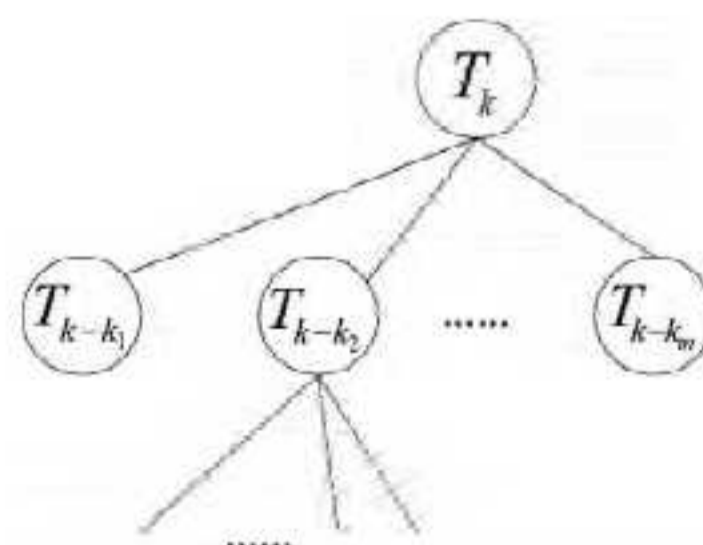


图 4 有界搜索树

有界搜索的过程是一个按深度优先搜索对有界搜索树进行遍历的过程,因此其时间复杂度与总节点的个数成正比。假设每一个父节点的分支至少有两个,则可证明树中的总节

点个数 n 不超过叶子节点 n_0 的 2 倍。

设 T 为以实例 (x, k) 为根节点的搜索树, T_k 表示 T 的叶子节点的数量, $T_{k-k_1}, T_{k-k_2}, \dots, T_{k-k_m}$ 为各子树的叶子节点数, 显然搜索树 T 的叶子节点数对应了递归式子: $T_k = T_{k-k_1} + T_{k-k_2} + \dots + T_{k-k_m}$ 。在假设与搜索树的每个叶子节点对应的实例足够简单并能够 $O(1)$ 求解的情况下, 搜索树 T 的叶子节点的精确数目与前述递归式子对应的特征多项式 $x^{k_m} = x^{k_m-k_1} + x^{k_m-k_2} + \dots + 1$ 的最大根 α 有关, 该类特征多项式的根可由有关数学工具直接求出, 则搜索树 T 的叶子节点数为 α^k , 因此分支限界搜索算法的时间复杂度为 $O(\alpha^k)$, α 一般是比 2 小比 1 大的实数。

分支限界排除了大量不可能的情况, 而且当对一个局部结构分支时相当于对参数进行了多重压缩。分支规则的正确性、高效性对问题的解决非常重要。当对问题有多种分解方法时, 应考虑哪个更高效。例如对最小顶点覆盖问题, 经过前文所提到的局部优化之后, 剩下的点的度最少是 3。假设我们以一个顶点 v 和它的邻居 $N(v)$ 作为局部结构来分支, 则要么把 v 算入最小顶点覆盖集里面, 要么将 $N(v)$ 算入, 则 $T_k \leq T_{k-1} + T_{k-3}$, 算法的复杂度为 $O(1.47^k)$; 如果从 $v, N(v), N(N(v))$ 这个局部结构来分支, 对于度大于 3 的顶点用刚刚的分支方法, 对于度最多为 3 的图, 在其满足 $N(N(v)) \geq 3$ 的二度点 v 处进行分支, 一个分支是将 $N(v)$ 放入解中, 参数值减少 2; 另一个分支是将 $v \cup N(N(v))$ 放入解中, 参数值至少减少 4。与前一条分支规则对应的递归函数为对应的 $T_k \leq T_{k-1} + T_{k-4}$, $\alpha = 1.381$, 后一条分支规则对应的是 $T_k \leq T_{k-2} + T_{k-4}$, $\alpha = 1.272$, 因此该分支的复杂度为 $O(1.39^k)$ 。

3.2.2 局部贪婪(greedy localization)

在参数计算领域, 局部贪婪技术是针对最大化问题的有效算法设计。总体说来, 局部贪婪技术研究的是问题的极大解和目标解 P_k (目标解可能不止一个) 的关系。具体说来, 它首先使用贪婪算法找到给定实例的一个极大解。假如该极大解已经达到了目标实例的要求, 则算法结束。否则, 获得的极大解应该与目标解有某种联系, 根据这额外获得的联系, 可以对原问题进行有效的求解——这就是所谓的“局部信息”。下面以 3-SET packing 问题^[42] (S 中包含有很多个集合, 每个集合有 3 个元素, 在 S 中寻找 k 个集合, 任意两个集合没有公共元素, 比如 P_3 问题——寻找最多的点不相交的的长度为 2 的路径问题) 为例, 来解释局部贪婪技术。

给定 3-Set Packing 问题的实例 (S, k) , 假设用贪婪算法可找到 S 中的一个极大 Packing W , 显然, $|W| < k$, 否则算法结束。假设目标解 $P_k = \{L_1, L_2, \dots, L_k\}$ 。不难看出, P_k 中的任意一个元组 $L_i (1 \leq i \leq k)$ 中的元素与 W 中的元素的交集不为空, 否则 W 还可以贪心得得到 L_i 这样一个 Packing, 与 W 是极大 Packing 相矛盾。因为 W 与目标解 P_k 中的任意元组都有公共元素, 穷举 $Val(W)$ ($Val(W)$ 表示极大 Packing W 中的所有元素) 中所有大小为 k 的子集。对于每一次枚举的大小为 k 的子集 P , 构造 k 个偏元组集合 P' , 其中每个偏元组包含 P 中一个元素和两个“*”。然后利用贪婪法将偏元组转换成正规元组 (在非 * 元素附近贪婪寻找)。如果正规元组的个数正好为 k , 算法返回 1, 否则转换过程新增的元素中必存在至少一个元素包含在目标解 P_k 中, 再通过枚举找到相应元素及其对应的元组, 这样就把 $Val(P')$ 中的“*”数目减一。

重复上述过程直到 P' 中的元组全部被转换为正规元组^[43]。

3.2.3 彩色编码(color coding)

彩色编码最初由 Alon 等人在 1995 年首次提出^[44], 它尤其适合应用于 k 个顶点集合构成的某种形状, 例如 k -path 问题 (在图中找到一条包含 k 个顶点的简单路径)、 k -cycle 问题 (在图中找一个包含 k 条边的圈) 等。通过利用 k 种不同颜色对顶点进行着色, k -path 问题转化为能否找到其顶点包含 k 种不同颜色的一条路径, 从而将节点不重复的限制转化为颜色不重复的限制, 达到简化问题的目的。对顶点的着色可以是随机着色, 也可以利用完美哈希函数族和全集来确定性地着色^[45]。颜色集的数目可以是 k , 也可以比 k 大。

将图中的 n 个节点随机着为 k 种颜色, 正确解 (如果存在) 所包含的 k 个节点一共有 k^k 种可能的着色情况。其中有 $k!$ 种情况被正确着色, 即此时 k 个节点的颜色各不相同。因此, 正确着色的概率就是 $k!/k^k$ 。利用 Stirling 近似公式 $k! \approx \sqrt{2\pi k} (k/e)^k$ 可知, 一次着色正确的概率大于 e^{-k} 。通过重复足够多的次数来保证获得的解具有一定的正确率, 在实际求解的过程中, 通常着色次数取 $O(e^k)$ 。着色完成之后, 常常利用动态技术等规划技术来寻找正确的解。当前对确定性彩色编码技术的研究主要集中在完美哈希函数族上, 相信人们在这方面能取得更大的突破。

3.3 其他主要技术

参数计算中, 其他主要技术还有随机核心、随机分治、measure and conquer、动态规划、图子式、整数规划等。其中图子式技术是较新的热门的技术。

20 世纪 80 年代发展起来的图子式理论 (graph minor)^[46] 重新焕发出活力, 为参数算法的研究提供了有力的支持^[47]。图子式理论从一开始就与算法理论密切相关^[48], 并对算法研究产生了深远的影响。一方面图子式理论发展出一些有效的算法设计技术, 其中图的树宽 (tree width) 和图的树分解 (tree decomposition) 是图子式理论中引入的两个重要概念; 另一方面借助图子式理论可以用一种非构造的方式从理论上证明某些问题的参数算法的存在性。图的树宽和树分解的引入给算法的分析和设计带来了一些新的思路。若将某些 NP 问题限定在有限树宽的图中, 则其求解是可行的。也就是说, 通过对图进行树分解, 若图的树宽是有限的, 则存在有效的多项式求解算法, 这对于很多场合的实际应用问题有着重要的意义。将图分解成若干部分, 然后在每部分上独立求解, 最后再结合起来, 是图子式常见的使用方法。

结束语 自从参数计算理论被提出以来, 人们已经提出了一系列的参数算法设计与分析技术, 并利用相应参数算法设计技术对许多领域的难解问题设计了有效的求解算法。尽管近年来参数计算理论在国内外的研究已经成为热点, 但是参数计算理论现正处于一个发展阶段, 还需要进一步丰富和发展。另外, 参数计算理论领域还存在诸多挑战, 例如新的算法设计与分析技术、核心化下界理论、参数算法工程实现、应用领域的拓宽等等。可进一步值得研究的问题具体为^[30] 核心化技术, 核心化技术是参数算法设计领域中一种最基本的算法设计技术, 一直以来是参数计算领域的研究热点之一。针对 FPT 问题, 参数理论有两个追求的目标: 一是想方设法设计出时间效率更高的算法; 二是想方设法找出更小的问题核。对这两个努力目标来说, 核心化技术都是最为基本的。

(1) FPT 问题核的下界研究; FPT 问题核的下界研究是参数理论在最近几年才开始的一个新的研究方向,核的下界研究无论在理论方面还是应用方面都有实际的指导意义。我们总是希望 FPT 问题经过核心化处理后可以得到多项式核甚至线性核,但是,并不是所有的 FPT 问题均存在多项式核。(2) FPT 问题证明及其核优化的研究;目前,依然存在许多的问题尚未被证明是否是固定参数可解的,如多项式识别测试问题、以树宽为参数的图同构问题等,对于未知其参数复杂性的问题,若我们能够应用核心化技术找到其问题核,则说明该问题是固定参数可解的。此外,针对 FPT 问题,进一步设计出问题核更小的算法,是参数理论的努力目标。(3) 核心化技术在复杂网络等领域的应用研究;核化技术目前主要应用于确定性问题的研究,从图论的角度来说,一般都是研究具有明确拓扑结构的图。许多实际问题,如生物分子相互作用网在不同的细胞或同一细胞的不同生活时刻是不完全相同的,而互联网的拓扑结构时刻也在动态变化之中,这些问题都具有不同程度的随机性和不确定性。

一方面我们需要扩展现有各种参数化技术的应用领域,另一方面是根据实际应用背景,设计特定情况下的参数化技术。

参 考 文 献

- [1] Graey M R, Johnson D S. Computers and Intractability: A Guide to the Theory of NP-Completeness[M]. New York: W. H. Freeman and Company, 1979
- [2] Chen Jia-ner. Parameterized computation and complexity: a new approach dealing with NP-hardness[J]. Journal of Computer Science and Technology, 2005, 20: 18-37
- [3] Ausiello G, Cresenzi P, Gambosi G, et al. Complexity and Approximation[M]. Springer-Verlag, 1999
- [4] Hochbaum D S. Approximation algorithms for NP-hard problems[M]. Boston, MA: PWS Publishing Company, 1997
- [5] Pearl J. Heuristics[M]. Addison-Wesley Pub, 1984
- [6] Michalewicz Z, Fogel D B. How to Solve It: Modern Heuristics [M]. Springer-Verlag, 2000
- [7] Motwani R, Raghavan P. Randomized Algorithms [M]. Cambridge University Press, 1995
- [8] Mitzenmacher M, Upfal E. Probability and Computing: Randomized Algorithms and probabilistic Analysis [M]. New York (NY): Cambridge University Press, 2005
- [9] Downey R G, Fellows M R. Parameterized Complexity [M]. New York: Springer, 1999
- [10] Roth-Korostensky C. Algorithms for Building Multiple Sequence Alignments and Evolutionary Trees[D]. ETH Zurich, 2000
- [11] Stege U. Resolving Conflicts from Problems in Computational Biology[D]. ETH Zurich, 2000
- [12] Chen J, Kanj I A, Jia W. Vertex cover: further observations and further improvement[J]. Journal of Algorithms, 2001, 41: 280-301
- [13] Gottlob G, Scarcello F, Sideri M. Fixed-parameter complexity in AI and non-monotonic reasoning [J]. Artificial Intelligence, 2002, 138: 55-86
- [14] Scott J. Efficient algorithms for detecting signaling pathways in protein interaction networks[J]. Journal of Computational Biology, 2006, 13: 133-144
- [15] Lindner C, Rothe J. Fixed-parameter tractability and parameterized complexity applied to problems from computational social choice[M] // Holder A, ed. Mathematical Programming Glossary, 1996
- [16] Betzler N, Guo J, Niedermeier R. Parameterized computational complexity of Dodgson and Young elections[J]. Information and Computation, 2010, 208(2): 165-177
- [17] Grohe M. The parameterized complexity of database queries [C] // Paredaens J, Su Jian-wen, eds. Proc. of the 20th ACM Symp. on Principles of Database Systems. ACM Press New York, NY, USA, 2001: 82-92
- [18] Wang J, Wang J, Chen J, et al. An automated signature generation approach for polymorphic worm based on color coding[C] // Proceedings of IEEE International Conference on Communications, 2009: 1-6
- [19] Nesetril J, Poljak S. On the complexity of the subgraph problem [C] // Commentationes Mathematicae Universitatis Carolinae, 1985, 26: 415-419
- [20] Fernau H. Edge dominating set: efficient enumeration-based exact algorithms[C] // Bodlaender H L, Langston M, eds. Proc. of 2nd International Workshop on Parameterized and Exact Computation. Berlin: Springer-Verlag, 2006: 142-153
- [21] Fernau H. Parameterized algorithms for hitting set: the weighted case[C] // Calamoneri T, Finocchi I, Italiano G F, eds. Proc. of the 6th Italian Conference on Algorithms and Complexity. Berlin: Springer-Verlag, 2006: 32-343
- [22] Abu-Khzam F. Kernelization algorithms for d-hitting set problems[C] // Dehne F K H A, Sack J-R, Zeh N, eds. Proc. of the 10th Workshop on Algorithms and Data Structures. Berlin: Springer-Verlag, 2007: 434-445
- [23] Bousquet N, Daligault J, Thomasse S. Multicut is FPT [C] // Proceeding STOC' 11 Proceedings of the 43rd Annual ACM Symposium on Theory of Computing, 2011: 459-468
- [24] Downey R, Fellows M R. Fundamentals of Parameterized complexity[M]. Springer, 2013
- [25] Chen J, Friesen D, Jia W, et al. Using nondeterminism to design efficient deterministic algorithms[J]. Algorithmica, 2004, 40: 83-97
- [26] Sloper C, Telle J A. An overview of techniques for designing parameterized algorithms[J]. Theory Computer Journal, 2008, 51: 122-136
- [27] Prieto E, Sloper C. Looking at the stars [J]. Theoretical Computer Science, 2006, 351: 437-445
- [28] Huffner F, Niedermeier R, Wernicke S. Techniques for practical fixed-parameter algorithms[J]. The Computer Journal, 2008, 51(1): 7-25
- [29] 刘运龙. 若干 NP 难解问题的参数化算法研究[D]. 长沙: 中南大学, 2009
- [30] 李绍华, 王建新, 冯启龙, 等. 参数计算中核心化技术[J]. Journal of Software, 2009, 20(9): 2307-2319
- [31] Chen Jia-ner, Kanj I A, Xia Ge. Simplicity is Beauty: Improved Upper Bounds for Vertex cover[OL]. <http://wenku.baidu.com/view/22ae88dbcezf0066f53322aa.html>
- [32] Abu-Khzam F N, Collins R L, Fellows M R, et al. Kernelization algorithms for the vertex cover problem: theory and experiments [C] // Proceedings of Workshop on Algorithm Engineering and Experiments (ALENEX), 2004
- [33] Nemhauser G L, Trotter L E. Vertex Packings: Structural properties and algorithms[J]. Math. Programming, 1975(8): 232-

- [34] Fellows M. Blow-ups, win/win's and crown rules: some new directions in FPT [C] // LNCS 2880. Berlin: Springer-Verlag, 2003: 1-12
- [35] Chlebik M, Chlebikova J. Crown reductions for the Minimum Weighted Vertex cover problem [J]. Electronic Colloquium on Computational Complexity, 2004, 101
- [36] Mathieson L, Prieto E, Shaw P. Packing edge disjoint triangles: A parameterized view [C] // LNCS 3162. 2004: 127-137
- [37] Ning D. Parameterized algorithm for the matching and packing problem [D]. Changsha: Central South University, 2007
- [38] Reed B, Smith K, Vetta A. Finding odd cycle transversals [C] // Operation. Research. Letter, 2004, 32(4): 299-301
- [39] 王建新, 江国红, 李文军, 等. 反馈集问题的研究进展 [J]. Computer Science, 2011, 38(1)
- [40] Guo Jiong, Gramm J, Huffner F, et al. Improved fixed parameter algorithms for two feedback set problems [C] // LNCS 3608, WADS 2005. Berlin: Springer, 2005: 158-168
- [41] Dehne F, Fellows M, Langston M, et al. An fast fpt algorithm for the undirected feedback vertex set problem [J]. Theory Computer System, 2007, 41(3): 479-492
- [42] Liu Y, Lu S, Chen J, et al. Greedy localization and color-coding: improved matching and packing algorithms [C] // Bodlaender H L, Langston M A. eds. Proc. of the 2nd International Workshop on Parameterized and Exact Computation. Berlin: Springer-Verlag, 2006: 84-95
- [43] 冯启龙. packing 和 matching 问题的参数算法研究 [D]. 长沙: 中南大学, 2010
- [44] Alon N, Yuster R, Zwick U. Color-coding [J]. Journal of the ACM, 1995, 42: 844-856
- [45] 王建新, 陈志彪, 陈建二. 最长路径问题研究进展 [J]. 计算机科学, 2009, 36(12)
- [46] Robertson N, Seymour P D. Graph Minors. II. Algorithmic Aspects of Tree-width [J]. Journal of algorithms, 1986, 7(3): 309-322
- [47] 高文字, 李邵华. 图的树分解及其算法应用研究进展. 计算机科学, 2012, 39(3)
- [48] Lu J H, Ling T B, Bao Z F, et al. Extended XML tree matching: theories and algorithms [J]. IEEE Trans on knowledge and data engineering, 2011, 23(3): 402-416

(上接第 17 页)

- [25] Aalst W, Hofstede Ter A H M. Workflow patterns put into context [J]. Software System Model, 2012, 11(3): 319-323
- [26] The YAWL Foundation. YAWL-User Manual Version 2. 3 [R]. 2012
- [27] White S, Miers D. BPMN Modeling and Reference Guide: Understanding and Using BPMN [M]. Lighthouse Point, FL, USA: Future Strategies, 2008
- [28] Chinosi M, Trombetta A. BPMN: an introduction to the standard [J]. Computer Standards & Interfaces, 2012, 34(1): 124-134
- [29] OMG. formal/2011-01-03 Business Process Model and Notation (BPMN) Version 2. 0 [S]. Needham, MA, USA: Object Management Group, 2011
- [30] Salatino M. jBPM Developer Guide [M]. Birmingham, UK: Packt Publishing, 2009
- [31] JBoss jBPM jPDL 3. 2 [OL]. <http://docs.jboss.com/jbpm/v3/user-guide/>, 2012
- [32] Chinosi M. Representing business processes: conceptual model and design methodology [D]. Varese, Italy: University of Insubria, 2009
- [33] OASIS. Web services business process execution language [OL]. <http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html>, 2007
- [34] WfMC. WfMC-TC-1025 Process Definition Interface--XML Process definition Language Version 2. 2 [S]. Workflow Management Coalition, 2012
- [35] Murata T. Petri nets: Properties, analysis and applications [J]. Proceedings of the IEEE, 1989, 77(4): 541-580
- [36] Xiong P, Fan Y, Zhou M. A petri net approach to analysis and composition of web services [J]. IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans, 2010, 40(2): 376-387
- [37] Li Z W, Hu H S, Wang A R. Design of liveness-enforcing supervisors for flexible manufacturing systems using petri nets [J]. IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, 2007, 37(4): 517-526
- [38] Zapletal M, van der Aalst W, Russell N, et al. An analysis of Windows workflow's control-flow expressiveness [C] // Proceedings of the 2009 7th IEEE European Conference on Web Services. 2009: 200-209
- [39] Peng L, Zhou B. Research on workflow patterns based on jBPM and jPDL [C] // Proceedings of the Pacific-Asia Workshop on Computational Intelligence and Industrial Application, 2008. 2008: 838-843
- [40] Ottensooser A, Fekete A, Reijers H A, et al. Making sense of business process descriptions: an experimental comparison of graphical and textual notations [J]. Journal of Systems and Software, 2012, 85(3): 596-606
- [41] Wang J. Emergency healthcare workflow modeling and timeliness analysis [J]. IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans, 2012, 42(6): 1323-1331
- [42] Wang H, Rutle A, MacCaul W. A formal diagrammatic approach to timed workflow modelling [C] // Proceedings of the 6th International Symposium on Theoretical Aspects of Software Engineering. 2012: 167-174
- [43] Muehlen M, Indulska M. Modeling languages for business processes and business rules: a representational analysis [J]. Information Systems, 2010, 35(4): 379-390
- [44] Tka M, Ghannouchi S A. Comparison of business process models as part of BPR projects [J]. Procedia Technology, 2012, 5: 427-436
- [45] Ehrig M, Koschmider A, Oberweis A. Measuring similarity between semantic business process models [C] // Proceedings of the 4th Asia-Pacific Conference on Conceptual Modelling. 2007: 71-80
- [46] Dongen B, Dijkman R, Mendling J. Measuring similarity between business process models [C] // Proceedings of the 20th International Conference on Advanced Information Systems Engineering. 2008: 450-464
- [47] Zhuo H H, Yang qiang, Hu D H, et al. Learning complex action models with quantifiers and logical implications [J]. Artificial Intelligence, 2010, 174(18): 1540-1569
- [48] Nau D, Ghallab M, Traverso P. Automated Planning: Theory & Practice [M]. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2004