

初等组合最优化问题的算法的发现 - 2000年10月

秦裕媛 · 14 Jul 2025

主编: 许国志 副主编 顾基发 车宏安
<系统科学与工程研究> ISBN 7-5428-2430-9
第277-293页
上海科技教育出版社2000.10

摘要 根据以下三个题目:第一, 讨论组合最优化的定义(2.1); 第二, 讨论最优化原理, “整体最优解是局部最优的”; 第三, 把导数差分的思想移植过来, 建立了对称差分解法(5.3-4)。将所得的方法汇集成一个树状结构, 提出一个求解初等组合最优化问题的方法的方法。具体到某个问题, 希望能依此发现它的算法。

关键词: 组合最优化定义;三种类型的最优化原理;对称差分解法。

AMS(1991)主题分类:90C27。

1 引言

组合最优化是一门新兴的、发展迅速的学科, 是数学规划和运筹学的一个分支。

研究组合最优化的问题确实需要[1, 第2页] “新的方法、新的探索途径和新的数学上的顿悟”(new methods, new approaches and new mathematical insights)。经典的思维和方法对本学科究竟能够起到多大的作用, 依然是值得深思的。作者一直[6-18]在探讨三个题目:

第一, 组合最优化的定义是什么?能够从它得到什么启迪?

第二, 我们所研究的是最优化问题, 当然应该弄清楚最优化原理 “整体最优解是局部最优的, 如果它在局部。”

第三, 如何把导数和差分的思想移植到本学科中来, 为我所用?

本文将特别研究如何从这三者得到求解问题的方法, 汇集成一个树状结构, 提出一个求解初等组合最优化问题的方法的方法。具体到某个问题, 就可依此发现它的算法。

2 组合最优化的定义

2.1 组合最优化的定义

迄今没有一个公认的定义。本文接受马仲蕃[4]所给的定义如下：

组合最优化是在给定有限集的所有具备某些特性的子集中，按某种目标找出一个最优的子集的一类数学规划。

用 π -ABC 或 $ABC \in \pi$ 表示具有一组性质 π 的术语ABC。根据上述，我们作如下的定义

定义 2.1 组合最优化是一类数学规划。

一个组合最优化问题，记作 $Problem\ XYZ$ ，是指对于每一个 $\pi^{(1)}$ -集合 S ，在其所有的 $\pi^{(2)}$ -(子)集合中求一个 $\pi^{(3)}$ -(优化)集合。

在给定的 $\pi^{(1)}$ - S 中寻找一个 $\pi^{(3)}$ -(优化)集合的任务叫做问题的一个实例，记作 $Instance\ XYZ: S$ ，或简记作 $Instance\ S$ 。

一个问题是由它的所有实例所组成。为了探讨一个问题的理论方面的事情，我们总是选一个具有代表性的实例，即所谓的典型实例入手。

对于 $Instance\ XYZ: S$ ，把 $\pi^{(1)}$ -集合 S 叫做论域。

S 的 $\pi^{(2)}$ -(子)集合叫做可行解或可行集。可行集的元素叫做它的可行元素，而在 S 中但不属于该可行集的元素叫做关于该可行集的自由元素。

可行集簇叫做可行域（解，集），记作 $\mathbf{D}$ 。

S 的一个子集合叫做碎片(fragment)，如果它属于某个可行集。碎片簇记作 $\mathbf{R}$ 。

S 的一个子集合或者一个可行集可以对应一个或者几个值。如果有两个，把一个叫做(主)值，另一个叫做参数。可行集 a 的主值记作 $g(a)$ ，参数值记作 $f(a)$ ，定义在 $\mathbf{D}$ 上的 $g(a)$ 叫做目标函数。

实例的答案包括最优解，最优值以及与本问题有关的信息。

$Problem\ XYZ$ 的一个算法是一组规则，用来在有限多步内解出该问题的某类实例。

一类问题的一个方法是一组规则，用来在有限多步内直接导得类内每个问题的一个算法。

为了突出一个命题的重要性，或者便于引用，我们有时把这个命题叫做原理。

定义 2.2 组合最优化论是研究满足定义2.1的诸问题的理论、原理、方法、算法和应用的学问。

2.2 正则问题

定义 2.3 给定 Instance XYZ: S 在 $\mathbf{D}$ 上, 有一个目标函数 $g(a)$ 。对于任一 $\pi^{(1)}$ -集合 $A \subset S$, 在 Instance XYZ: A 的可行域 $\mathbf{D}'$ 上, $h(a)$ 是它的目标函数。如果对于 $a \in \mathbf{D} \cap \mathbf{D}'$, 总有 $g(a) = h(a)$, 把 Instance XYZ: S 叫做正则的。如果每一个实例是正则的, Problem XYZ 是正则的。

正则性可以把组合最优化问题分为两类。把它写成一个命题, 并用符号 $P_0^{(0)}$ 来表示。

$P_0^{(0)}$: Problem XYZ 是正则的。

给定 $P_0^{(0)}$ -Instance S , 每个 Instance $A (\subset S)$ 是它的一个子实例。除去空子实例和 Instance S 本身, 其它的子实例是真子实例。一个实例是简单的, 如果它没有真子实例。

众多的组合最优化问题都是正则的。但是, 例如, 决策论中的 Savage 的后悔法就不是[8]。

2.3 特性集 List PPP

本文还将列出关于 $\pi^{(i)} (i = 1, 2, 3)$ -集的基本性质和公理, 记作 $P_j^{(i)}$ 或 $P_{jk}^{(i)}$, 再用 $P_{jk}^{(4)}$ 记作最优化原理的特性。可能还有什么别的特性, 用增加上标总的数字加以记载, 把它们汇集在一起 构成特性集, 记作 List PPP (**p**roperties, **p**ostulates and **p**inciples of optimality)。

集合的序关系是重要的。我们列出以下两个对于 $\pi^{(1)}$ -集合 (论域) 的特性,

$P_{01}^{(1)}$: $\pi^{(1)}$ -集合是半序的。

$P_{02}^{(1)}$: $\pi^{(1)}$ -集合是全序的。

取子集合该是最基本的一种运算。一个圈的任一真子图都不再是圈。一个连通图的真子图可能是连通的, 也可能不是。而一个简单图的任何真子图总是简单的。这些表明取子集的情形是复杂的。但是, 一个 $\pi^{(1)}$ -集合 S 的子集合也就是这么三个可能性。接受这一事实, 有助于引用。把后两种写成特性:

$P_{11}^{(1)}$: 有的 $\pi^{(1)}$ -集合有 $\pi^{(1)}$ -真子集合。

$P_{12}^{(1)}$: $\pi^{(1)}$ -集合的每一个真子集合都是 $\pi^{(1)}$ -的。

对于 $\pi^{(2)}$ -集合 (可行域), 也有同样的特性:

$P_1^{(2)}$: 有的 $\pi^{(2)}$ -集合 a 有 $\pi^{(2)}$ -真子集合。

$P_{11}^{(2)}$: 对于每一个 $\pi^{(2)}$ -集合 A , 存在一个 $x \in A$ 使得 $Anx \in \pi^{(2)}$ 。

$P_2^{(2)}$: $\pi^{(2)}$ -集合 a 的每一个真子集合都是 $\pi^{(2)}$ 的。

以后还将有其它的命题添加到特性集中来。

2.4 枚举法和隐枚举法

从定义2.1可以立刻得到求解组合最优化问题的三个基本方法，它们是枚举法，隐枚举法和同解法。本节讲前两个。第一个是枚举法。

对于 *Instance* XYZ: $P = \{p_j : 1 \leq j \leq n\}$ ，有 $P = p_1 \otimes p_2 \otimes \dots \otimes p_n$ 。一个系统搜索 P 的所有子集合的公式是

$$\prod_n = \prod_{\otimes} \{(e \oplus p_j) : 1 \leq j \leq n\}$$

其中 $\otimes$ 是集合 P 中的形式乘法， e 是单位元素， $\oplus$ 是一个形式加法。这两个运算满足交换律，结合律和分配律。它的展开式的每一项是 P 的一个子集合。先从中挑选所有的可行解，再按它们的值挑出最优的。这就是枚举法，记作 *Method* ENUMERATE。

一般而言，直接从定义2.1来求解问题都难有效，但是它非常直观，按章办事，此其一。其次，计算机作为极为重要的计算工具，虽然它的巨大发展使得它有飞速计算的能力和海量存储的空间，但是节约求解费用(计算量)依然是一个科学原则。最后，在数学甚至每一个科学领域里，人们总是在追求有效性。计算 n 阶行列式的值，求解 n 阶线性方程组，计算函数的导数等等，人们都是从发展理论入手，再发现有效方法。在组合最优化论中对各种问题寻求有效算法也是一个中心课题。

作为一个例子，讨论 $\{P_2^{(2)}, P_{12}^{(3)}\}$ -问题，其中 $P_{12}^{(3)}$ 是指目标函数等于可行集的诸元素的值之和， $P_2^{(2)}$ 等价于：如果集合 A 有一个子集不是 $\pi^{(2)}$ -的，则 A 不是 $\pi^{(2)}$ 的。应用它，并规定

$$(i) \prod_0^\wedge \equiv e。$$

$$(ii) \prod_{j+1}^\wedge \text{ 等于从 } (\prod_{j+1}^\wedge) \otimes (e \oplus p_{j+1}) \text{ 删去所有非 } \pi^{(2)}\text{-集合的结果。}$$

于是得到两个方法，都叫做隐枚举法，记作 *Method* IMPL-ENUM- x ($x = 1, 2$)。一个是先列出所有的可行解，再从(主)值求出最优值和所有的最优解；另一个是求得每一个碎片时，计算它的值，然后得到答案。

2.5 第三个基本方法 同解法

对于 *Problem* XYZ, 设有两个实例 *Instance* S 和 *Instance* T ，如果它们有相同的答案，就写作 $Instance\ S = Instance\ T$

改进规则是指能断定所给实例是否可改进，并能完成相应任务的规则。对于 *Instance* S ,

(i) 如果可改进，就能构造一个改进实例 *Instance* T ；即与 *Instance* S 具有相同答案，且按所规定的准则，比 *Instance* S 更“好”的实例。

(ii) 如果不可改进，能够断定集合 S 本身就是 *Instance* S 的一个最优解。

对于(i)，我们写 *Instance* $T \preceq Instance\ S$ ，或者简记作 $T \preceq S$ 。两者之间的关系可以写成 $T = \rho(S)$ ，其中 ρ 叫做改进子(improvor)。

对于(ii), 如果Instance S 不可改进, 我们有 $\rho(S) = S$ 。

给定Instance XYZ: S , 如果有一个改进子 ρ 使得

$$S \equiv S^0, S^1 = \rho(S^0), S^{(i+1)} = \rho(S^{(i)}), i = 1, 2, \dots$$

而且存在一个正整数 k 有 $S^{(k)} = \rho(S^{(k)})$, 那么所给实例有一个最优解 $S^{(k)}$ 。改进子 ρ 提供的方法叫做第一同解法。

一般地, 如果有一个改进子序列 $\rho^{(1)}, \rho^{(2)}, \dots, \rho^{(k)}, \dots$, 且有

$$S^{(i+1)} = \rho^{(i+1)}(S^{(i)})$$

这个改进子序列提供的方法叫做第二同解法。

有时, 改进规则还要求如下的规则。

(iii) 设有正整数 $t(\leq +\infty)$ 。如果 $S^{(t+i)} \neq S^{(t)}$, 则断定实例(在 t 步之内)没有最优解。

至此, 我们从定义得到了三组五个求解方法。今后还要从三个最优化原理和关于可行集的对称差导出许多方法。为了记述这些方法成为一个有规律的集合, 叫做方法集。我们像特性集那样, 采用符号 $M_{jk}^{(i)}$ 表示第 i 组第 j 个方法的第 k 变形。 $i = 0$ 表示从定义2.1直接得到的方法; 当 $i = 1, 2, 3$ 时, 依次为从下面要讲的第一, 二, 三最优化原理得到的方法; $i = 4$ 时是从对称差导得的方法。还有近似法, 启发法, 随机法等, 需要继续编号。因此, 枚举法记作 $M_1^{(0)}$, 两个隐枚举法记作 $M_{21}^{(0)}$ 和 $M_{22}^{(0)}$, 两个同解法记作 $M_{31}^{(0)}$ 和 $M_{32}^{(0)}$ 。

2.6 初等组合最优化论

人们熟知, 组合最优化中有一个非常重要的理论部分, 就是计算复杂性。讨论了多项式(P)问题, NP问题, NP完全问题以及求解方法等等。我们规定

定义 2.4 初等组合最优化论讨论组合最优化论中的正则问题, 且不涉及NP概念和凸多面体理论的部分。

3 第一（论域型）最优化原理

3.1 第一（论域型）最优化原理

下面是我们关心的一般最优化原理：

$P^{(4)}$ ：最优化原理 整体最优解在局部是最优的。

在组合最优化中, 给定Instance XYZ: S 时, 它有三种集合(或簇):论域 S , 可行域 $\mathbf{D}$ 和某个可行集 a 。我们自然希望把这个一般原理应用于它们, 建立三个相应的原理。

首先，在论域 S 上，把(子)集合 D 作为整体，其子集合 A 为局部。如果 *Instance A*及*Instance D*都有解，那么一般原理可以写成如下的命题。

$P_1^{(4)}$ ：第一（论域型）最优化原理 *Instance XYZ*: S 的论域是 $S, A \subseteq D \subseteq S$ ， 那么整体 D 的最优解是局部 A 的最优解，如果它在局部之中。

不失一般性，只讨论适定性实例。 即对于 $A(\subseteq S) \in \pi^{(1)}$ ， *Instance XYZ*: A 的解总是唯一的，记作 A^* ；当 A 非空，它的解 A^* 也是非空的。

若 $A, B \subseteq S$ ， *Instance A*， *Instance (A ∪ B)*及*Instance (A* ∪ B)*分别有(唯一)最优 A^* ， $(A \cup B)^*$ 及 $(A^* \cup B)^*$ ， 把 A ， $A \cup B$ 分别作为局部与整体， $P_1^{(4)}$ 可以写成

若 $(A \cup B)^* \subseteq A \subset A \cup B$ ， 则 $(A \cup B)^* = A^*$ 。

等价地
$$(A^* \cup B)^* = (A \cup B)^*$$

于是 $P_1^{(4)}$ 可以写成一个便于代数演算的形式：

$P_1^{(4)}$ ：第一（论域型）最优化原理（公理形式） 在*Instance XYZ*: S 中， 设实例 $A \subseteq S$ 的最优解记作 A^* ， 则

$P_{11}^{(4)}$ ：唯一性， $A(\in \pi^{(1)})$ 有唯一的优化集合 A^* ， 写作 $A = A^* \cup \bar{A}$ ， $A^* \cap \bar{A} = \emptyset$ 。

$P_{12}^{(4)}$ ：存在性， $A^* \neq \emptyset$ 当且仅当 $A \neq \emptyset$ 。

$P_{13}^{(4)}$ ：对于所有的 A, B 使得 $A, A \cup B, A^* \cup B \in \pi^{(1)}$ ， 总有 $(A^* \cup B)^* = (A \cup B)^*$ 。

3.2 四个求解的方法

去劣(自由元素)法 $M_1^{(1)}$ ：把所有 $\{P_1^{(1)}, P_1^{(4)}\}$ -问题构成 $\{P_1^{(1)}, P_1^{(4)}\}$ 问题类。 对于实例 $\{P_1^{(1)}, P_1^{(4)}\}$ -*Instance XYZ*: S ， 由 $P_{13}^{(4)}$ 可以求得求解实例的两个方法。 第一个规则记作DELETE， 是从这等式的右端到左端， 得到去劣(自由元素)法， 记作 $M_1^{(1)}$ 。 设想有*Instance XYZ*: D ， 如果能够把 D 写为 $A \cup B$ ， 而且能够容易求出 A^* ， 即从 A 中删去若干个关于 A^* 的自由元素。 $A \cup B$ 与基数较小的 $A^* \cup B$ 是同解的。 如果能够不断地做下去， 就能得到最优解。

扩展法 $M_2^{(1)}$ ：应用 $P_{13}^{(4)}$ 的第二个规则， 记作EXTENSION， 是从等式的左端到右端：如果 A 的优化集合是 A^* ， 增添了集合 B ， 还能求出 $(A^* \cup B)^*$ ， 就等于求出了 $(A \cup B)^*$ 。 这个思路得到扩展法 $M_2^{(1)}$ 。

定理 3.1 一个问题可以用去劣法和扩展法求解， 当且仅当它属于 $\{P_1^{(1)}, P_1^{(4)}\}$ -问题类。

生成法 $M_3^{(1)}$ ：设实例 $\{P_2^{(1)}, P_2^{(2)}, P_1^{(4)}\}$ -*Instance XYZ*: $S = (P, \dots)$ 的主集合 P 是 $\{p_j : 1 \leq j \leq n\}$ 。 在 $M_2^{(1)}$ 中， 令 $B_j = \{p_j\}, 1 \leq j \leq n$ ， 就得到一个方法， 叫做生成法 $M_3^{(1)}$ 。

分治法 $M_4^{(1)}$ ：对于 $1 \leq s \leq n - 1$ ， 令 $A_0 = \emptyset$ ， $A_{s+1} = A_s \oplus \{p_{s+1}\}$ ， $A_0^* = \emptyset$ 及

$$A_{s+1}^* = A_s^* \overline{\oplus} \{p_{s+1}\} = (A_s^* \oplus \{p_{s+1}\})^*$$

如果写 $A_{ij} \equiv A_i \oplus A_{i+1} \oplus \dots \oplus A_j$, 及 $A_{ij}^* \equiv A_i \oplus \bar{A}_{i+1} \oplus \dots \oplus \bar{A}_j$, 则有

$$\begin{aligned} S^* &= A_1 \oplus \bar{A}_2 \oplus \bar{A}_3 \oplus \bar{A}_4 \oplus \dots \oplus \bar{A}_{2^n} = A_{12} \oplus \bar{A}_{34} \oplus \bar{A}_{56} \oplus \bar{A}_{78} \oplus \dots \oplus \bar{A}_{2^{n-1}, 2^n} \\ &= A_{14} \oplus \bar{A}_{58} \oplus \dots \oplus \bar{A}_{2^{n-3}, 2^n} = \dots = A_{1, 2^{n-1}} \oplus \bar{A}_{2^{n-1}+1, 2^n} = A_{1, 2^n} \end{aligned}$$

这就得到了分治法 $M_4^{(1)}$ 。

3.3 几个例

例 3.1 平面凸壳问题, 最小支撑树问题, 首 k 个最小数问题(在有限个实数中求最小的 k 个数)等都满足第一最优化原理, 上述四个方法 $M_j^{(1)} (j = 1, 2, 3, 4)$ 也都能应用, 并可以建立相应的算法以及它们的变形。

例 3.2 二部分图中值和最大匹配问题不满足第一最优化原理。 例如有一个赋值二部分图 $G = (V_1, V_2, E; r)$, 其中 $V_1 = \{v_1, v_3\}, V_2 = \{v_2, v_4\}$, 边集合和边的值是 $(E : r) = \{v_1 v_2 : 2, v_2 v_3 : 3, v_3 v_4 : 2\}$ 。 E 是主集合。

设 $A = \{v_1 v_2, v_2 v_3\}$ 及 $B = \{v_3 v_4\}$, 于是有 $A^* = \{v_2 v_3\}$, $(A^* \cup B)^* = \{v_2 v_3\}$, 但是 $(A \cup B)^* = \{v_1 v_2, v_3 v_4\}$ 。 所以不可能应用上面所列的四个方法得到相应的算法来求解二分图的值和最大匹配问题。

定理 3.2

(i) 二分图的值和最大匹配问题, 一般图最大匹配问题, 最大流问题, 以及最小费用最大流问题都没有从 $M_j^{(1)} (j = 1, 2, 3, 4)$ 得到相应的算法。

(ii) 中国邮路问题也没有从 $M_j^{(1)} (j = 1, 2, 3, 4)$ 得到相应的算法。

这是因为二分图的值和最大匹配问题是(i)中的其它问题的子问题。 因此所有这些问题都不满足第一最优化原理。 而中国邮路问题要用到完全图中的值和最大匹配问题的算法。

4 第二与第三最优化原理

4.1 第二（可行域型）最优化原理

关于可行域 $\mathbf{D}$, 取 $A \subseteq D \subseteq \mathbf{D}$, 有

$P_2^{(4)}$: 第二（可行域型）最优化原理（公理形式）

对于 $A \subseteq \mathbf{D}$, 设 $m = \text{opt}\{g(a) : a \in A\}$ 及 $A^* = \{a : g(a) = m, a \in A\}$ 。

$P_{21}^{(4)}$: 唯一性, A^* 是唯一的, 写作 $A = A^* \cup \bar{A}$, $A^* \cap \bar{A} = \emptyset$ 。

$P_{22}^{(4)}$: 存在性, $A^* \neq \emptyset$ 当且仅当 $A \neq \emptyset$ 。

$P_{23}^{(4)}$: $(A^* \cup B)^* = (A \cup B)^*$ 。

注意，这原理和第一最优化原理在形式上是相同的，但是算子 $*$ 的定义范围是不同的。

定理 3.3 每一个正则组合最优化问题具有第二最优化原理。

利用这个原理可以直接得到两个熟知的分支定界法，记作 $M_1^{(2)}$ 和 $M_2^{(2)}$ 。在这方面，我们没有什么新的见解，不赘说了。

4.2 第三（邻域型与碎片型）最优化原理与最优扩充问题

第三种最优化情形是关于可行解的碎片的。

碎片的子集合是碎片，后者是前者的子碎片，前者是后者的一个扩充。两者之差是后者的扩充部分。如果 $b^{(1)}$ 是一个碎片，它有一个扩充，记作 $a^{(1)} \cup b^{(1)} \equiv a^{(1)} \otimes b^{(1)}$ ，它的扩充部分是 $a^{(1)}$ ，记作 $b^{(1)}$ -扩充部分。在碎片簇 $\mathbf{R}$ 中，给定 $b^{(1)}$ 。把它的若干个扩充 $D = \{a^{(i)} \otimes b^{(1)} : 1 \leq i \leq t\}$ 作为整体。

$b^{(1)}$ -扩充部分 $A = \{a^{(i)} : 1 \leq i \leq t\}$ 作为局部。如果每个碎片都能定义一个主值，于是有

$P_3^{(4)}$ ：第三（邻域型与碎片型）最优化原理（公理形式）

若 $A, b^{(1)}, A \otimes b^{(1)} \subseteq \mathbf{R}$, $m = \text{opt}\{g(a) : a \in A\}$ 及 $A^* = \{a \in A : g(a) = m\}$, 则

$$(A^* \otimes b^{(1)})^* = (A^* \otimes b^{(1)})^*$$

在上面，我们用同一组符号 $\{\oplus, \otimes\}$ 表示了两种意义。一是集合间的运算 $A \oplus B = \{A, B\}$, $A \otimes B = A \cup B$ 。二是数量之间的运算。读者注意了上下文，阅读时将不致发生误解。

我们从 $P_3^{(4)}$ 得到

定理 3.4 $P_3^{(4)}$ 等价于下面的命题: 对于 $A, B, A \otimes B \subseteq \mathbf{R}$, 有 $(A^* \otimes B)^* = (A \otimes B)^*$ 。

定理 3.5 设 $P_{12}^{(3)}$ -Instance XYZ: $S = \{P; r\}$, 其中 $P = \{p_j : 1 \leq j \leq n\}$, 而 p_j 的值为 $r_j \in r$, 则实例有第三最优化原理。

满足上述定理的组合最优化问题叫做最优扩充问题。

例 4.1 最小支撑树问题

设有最小支撑树问题的一个实例 $P_{12}^{(3)}$ -Instance TREE: $G = (V, E, r)$, 其中 r 是一个实数集。

支撑树可以定义为主集合 E 的一个子集合 a , 如果它的导出支撑子图是边数最大的无圈支撑子图。每一个碎片 $b^{(1)}$ 对应一个支撑森林, 它的连通分支数为 $\omega(b^{(1)})$ 。碎片的值等于诸边的值之和。

对于一个碎片 $b^{(1)}$ ， 如果有一个最简单的真碎片(即一条边) $a^{(j)} \in E \setminus \{b^{(1)}\}$ ， 使得 $a^{(j)} \cup b^{(1)}$ 是另一个碎片， 边 $a^{(j)}$ 必然是连接 $b^{(1)}$ 的两个连通分支的边， 或者说 $\omega(b^{(1)}) = \omega(b^{(1)} \cup a^{(j)}) + 1$ 。 连接 $b^{(1)}$ 的任意两个连通分支的边构成集 $A = \{a^{(j)}\}$ 。 A 中最优碎片记作 A^* ， 它是值最小的边集合。 A^* 中任一条边添加到 $b^{(1)}$ ， 构成一个 $(\omega(b^{(1)}) - 1)$ -最优扩充。 于是有

定理 3.6 图 $G = (V, E, r)$ 中有一个森林(碎片) $\{(V_1, E_1), (V_2, E_2), \dots, (V_k, E_k)\}$ ， 其中 $G_i = (V_i, E_i)$ 是一个树。

如果 $e^* = v_{i_0} v_{j_0}$ 是满足

$$r(e^*) = \min\{r(v_i, v_j) : v_i \in E_i, v_j \notin E_i, 1 \leq i, j \leq k\}$$

的一条边， 则含有这个森林的(若干个)最小支撑树中至少有一个含有边 e^* 。

有的作者就把这定理作为建立最小支撑树问题的许多著名算法的出发点。

例4.2 最大独立集问题

有一个典型实例 $P_{12}^{(3)}$ -Instance INDPDT (P, I, r) ， 其中 $p_j (\in P)$ 的值是 $r_j (\in r)$ 。 实例的任务是求解值和最大的独立集。

给定一个碎片 I_0 ， 要求一个含有 I_0 的值最大的基集。 林诒勋[2]指出， 这是一个独立集最优扩充问题。 他讨论了这个问题并在排序论中得到了应用。

4.3 Bellman最优化原理与离散型动态规划

定义 4.1 数学系统 $\{S, \oplus, \otimes\}$ 叫做半环(semi-ring, SR)， 如果集合 S 是关于 $\oplus$ 的具有零元素 z 的可交换半群， 关于 $\otimes$ 的具有单位元素 e 的半群， 且分配律成立。

优选半环(optimizing semi-ring, OSR)是满足 $p \oplus q = p$ 或 $q, p, q \in S$ 的半环。 若 $p \oplus q = p (\neq q)$ ， 则说 p 不劣(优)于 q 。 如果 p 优(劣)于 e ， p 叫做阴(阳)元素， 而 e 是中性的。

强优选半环(strongly optimizing semi-ring, $SOSR$)是一个优选半环， 还满足条件:如果 $p \oplus q = p (\neq q)$ ， 则对于任一 $r (\neq z) \in S$ ， 有 $p \otimes r \oplus q \otimes r = p \otimes r (\neq q \otimes r)$ 。

这三种半环依次记作 $\{SR, \oplus, \otimes\}$ ， $\{OSR, \oplus, \otimes\}$ 和 $\{SOSR, \oplus, \otimes\}$ 。

设 $\overline{R} = R \cup \{+\infty\}$ ， $\underline{R} = R \cup \{-\infty\}$ ， $R_0^+ = R^+ \cup \{0\}$ 。 容易证明， $(\overline{R}, \wedge, +)$ (有的学者还叫做极小代数)， 是一个强优选半环。 它的零元素是 $+\infty$ ， 单位元素(即中性元素)是实数0。 负数是阴元素， 正数是阳元素。 同样， 可以证明， $(\underline{R}, \vee, +)$ (有的学者还叫做极大代数)， $(\overline{R}^+, \wedge, \times)$ 和 $(R_0^+, \vee, \times)$ 都是强优选半环。 $([0, 1], \wedge, \vee)$ 和 $([0, 1], \vee, \wedge)$ 是优选半环， 却不是强优选的。 $(R, +, \times)$ 可以作为半环的一个例子， 但不是优选的。

网络是赋值于 $\{SOSR, \oplus, \otimes\}$ 的无阴回路有向图。

最优路问题的一个实例， 记作 $P_{12}^{(3)}$ -Instance $(\underline{R}, \vee, +)$ -PATH: $N(V, L, r; u, v)$ ， 是在网络 $N(V, L, r)$ 中求解一条 (u, v) (即从顶点 u 到 v 的)-最优路。

路可以定义为主集合 L 的子集 a ，如果它的导出子图是一条途径(可以含有圈的图)，网络中没有值为阴的圈。因此只需要讨论路而不需要途径。把从 u 到 w 的路和从 w 到 v 的路作为可行解。它的每个子集作为碎片。要求最优路就是求一个值 $\otimes$ 积最优 (u, v) 可行解。所以最优路问题是一个组合最优化问题。

最简单的真碎片是一条边。取一条有向边 $b^{(1)} = us$ ，从 s 到 v 的一条路是 $b^{(1)}$ 的一个扩充部分。 $b^{(1)}$ 的最优扩充是所有的扩充部分 $A = \{a^{(j)}\}$ 中的最优集，记作 A^* ，值为 l_{uv} 。 A^* 中任一条路添加到 $b^{(1)}$ ，构成 $b^{(1)}$ 的一个最优扩充。从 u 出发的每一条有向边都有一个最优扩充。它们中的最优的是所求的一条 (u, v) -最优路。

从 u 出发取有向边 uw ，值为 r_{uw} 。它的最优可行扩充的值是 $r_{uw} \otimes l_{wv}$ 。取遍 u 的邻点 w ，有

$$l_{(uv)} = \text{opt}\{r_{uw} \otimes l_{wv} : w \in V\} = [r_{u1}, r_{u2}, \dots, r_{un}] \otimes [l_{1v}, l_{2v}, \dots, l_{nv}]^t \quad (4.3.1)$$

这是求解最优路问题的基本公式。我们有

定理 4.1 (Bellman最优化原理 1957) (u, v) -路 a^* 是最优的，当且仅当路上任取一点 s ，则在 a^* 上的 (s, v) -路是从 s 到 v 的所有路中最优的。

最优路问题是重要的，因为它们离散(固定与不固定阶段)型动态规划的几何模型。

4.4 多阶段有向图与嘉量原理

有向图 $D(V, L)$ 是 n (多)阶段的，如果顶点集 V 可以划分为这样的 $(n+1)$ 个部分 $V = Y\{V^{(i)} : 0 \leq i \leq n\}$ 及 $V^{(i)} = \{v_t^{(i)} : 1 \leq t \leq t_i\}$ ，使得一条有向边的终点在第 i 状态集合 $V^{(i)}$ 时，它的始点必然在第 $(i-1)$ 状态集合 $V^{(i-1)}$ 中。 $V^{(i-1)}$ 与 $V^{(i)}$ 的导出子图叫做第 i 阶段。

在 n 阶段赋嘉量有向图 $(SOSR, \oplus, \otimes)-D(V, L, r)$ 中，有向边 $v_s^{(i-1)}v_t^{(i)}$ 的(主)值叫做嘉量，记作 $J(v_s^{(i-1)}, v_t^{(i)}) = r_{st}^{(i)}$ 。若两个顶点之间没有有向边，就当作有一条嘉量为零元素 z 的有向边。于是第 i 阶段是一个单向的二分图，它的邻接矩阵是

$$STAGE(i) \equiv [STAGE(i)] \equiv [J(v_s^{(i-1)}, v_t^{(i)})] \equiv [r_{st}^{(i)}]$$

$D(V, L, r)$ 可写作

$$STAGE(1) \cup STAGE(2) \cup \dots \cup STAGE(n) \quad (4.4.1)$$

设

$$J(V^{(0)}, V^{(h)}) = [J_{ij}] = STAGE(1) \otimes STAGE(2) \otimes \dots \otimes STAGE(h) \quad (4.4.2)$$

结果是 $t_0 \times t_h$ 矩阵。 $J(V^{(0)}, V^{(h)})$ 的元素 J_{ij} 等于 $\sum_{\oplus} r_{ii_1}^{(1)} \otimes r_{i_1i_2}^{(2)} \otimes r_{i_2i_3}^{(3)} \otimes \dots \otimes r_{i_{n-1}j}^{(n)}$ ，其中 $r_{ii_1}^{(1)} \otimes r_{i_1i_2}^{(2)} \otimes r_{i_2i_3}^{(3)} \otimes \dots \otimes r_{i_{n-1}j}^{(n)}$ 是路 $v_i^{(0)}, v_{i_1}^{(1)}, v_{i_2}^{(2)}, \dots, v_{i_{n-1}}^{(n-1)}, v_j^{(n)}$ 的嘉量。 J_{ij} 是从 $v_i^{(0)}$ 到 $v_j^{(n)}$ 的最优路的嘉量。

定理 4.5 在(4.4.1)中，设 $V^{(0)} = \{v_0^{(0)}\}$ ， $V^{(n)} = \{v_0^{(n)}\}$ ，则

$$J(v_0^{(0)}, v_0^{(n)}) = STAGE(1) \otimes STAGE(2) \otimes \dots \otimes STAGE(n) \quad (4.4.3)$$

定理 4.6 (嘉量原理 1980)[5-7] 在强优选半环的网络上, (u, v) -路 a^* 是最优的, 当且仅当路上任取两点 s, t , 则在 a^* 上的 (s, t) -路是从 s 到 t 的所有路中最优的。

我们还开展了一系列的工作[5-7]。例如从(4.4.3)得到表格计算方法等等。

5 对称差分解(SDD)法

5.1 紧邻簇 $\mathbf{N}(a)$ 与改变度簇 $\mathbf{C}(a)$ 的分解

对于两个 $\pi^{(2)}$ -集合 a, b , 有两种差集。一是简单差, $a_b = a \setminus b$ 与 $b_a = b \setminus a$ 。把它们构成一个集合对 (a_b, b_a) 。另一是对称差 $a \Delta b = a_b \cup b_a$ 。这两种差的关系是

$$\tau((a_b, b_a)) \equiv \tau(a_b, b_a) = a_b \cup b_a$$

其中 $\tau(a_b, b_a)$ 叫做一个变换项。于是

$$b = a \Delta \tau(a_b, b_a) = (a \cup b_a) \setminus a_b \quad \text{与} \quad a = b \Delta \tau(a_b, b_a)$$

(a_b, b_a) 叫做 a 与 b 的互易对。互易对是简单的, 如果它没有真子对。

集合对 (U, V) 叫做互易(子)对, 如果 $U \subseteq a_b, V \subseteq b_a$, 而且变换项 $\tau(U, V) = U \cup V$ 满足条件 $a \Delta \tau(U, V) \in \pi^{(2)}$ 。

对于一个可行解 a , 考虑这样的简单集合对 (U, V) , $U \subseteq a, V \subseteq P \setminus a$ 使得 $b \equiv a \Delta \tau(U, V) \in \mathbf{D}$ 。 (U, V) 叫做关于 a 的一个简单互易对, 或者是 a 的一个增量(increment), b 是 a 的一个(1步)邻点(紧邻)。 U 是关于 a 的可行碎片, 而 V 是关于 a 的自由碎片。

给定赋值系统 $System S = (P, r)$, 每一个碎片 U 有一个(主)值, 有时还有一个参数值, 分别写作参数值: $f(U)$ 和主值 $g(U)$ 。对于所有 $U \in \mathbf{R}$, U 的一个参数 $f(U)$ 可能就是碎片 U 的基数或者别的什么。

a 的增量簇记作 $\mathbf{C}(a)$, a 的紧邻簇记作 $\mathbf{D}(a)$:

$$\mathbf{D} \equiv \mathbf{D}(a) = \{a \Delta (U, V) : (U, V) \in \mathbf{C}(a)\} \quad (5.1.1)$$

$$\text{令} \quad \mathbf{N}_{\Theta\Omega} = \mathbf{N}_{\Theta\Omega}(a) = \{b \in \mathbf{D}(a) : f(b) - f(a) \Theta 0, g(b) - g(a) \Omega 0\} \quad (5.1.2)$$

其中 $\Theta, \Omega \in \{<, =, >\}$ 。

$$\text{令} \quad \mathbf{N}_{\bullet} = \mathbf{N}_{\Theta<} \cup \mathbf{N}_{\Theta=} \cup \mathbf{N}_{\Theta>}$$

$$\text{及} \quad \mathbf{N}_{\bullet\Omega} = \mathbf{N}_{<\Omega} \cup \mathbf{N}_{=\Omega} \cup \mathbf{N}_{>\Omega}$$

就有下面的分解公式

$$\mathbf{D}(a) = \mathbf{N}_{<} \cup \mathbf{N}_{\bullet} \cup \mathbf{N}_{>} \quad (\text{参数形式}) \quad (5.1.3)$$

$$= \mathbf{N}_{.<} \cup \mathbf{N}_{.=} \cup \mathbf{N}_{.>} \text{ (主值形式)} \quad (5.1.4)$$

最简单又最重要的目标函数之一是

$$P_{12}^{(4)} : g(a) = \sum \{r_j : p_j \in a\}, \text{ (值和形式)} \quad (5.1.5)$$

而参数函数表示基数, (5.1.2) 等价于下面的公式。

$$\mathbf{C}_{\Theta\Omega} = \mathbf{C}_{\Theta\Omega}(a) = \{(U, V) : (U, V) \in \mathbf{C}(a) : f(V) - f(U) \Theta 0, g(V) - g(U) \Omega 0\}.$$

$$\text{令} \quad \mathbf{C}_{\Theta\bullet} = \mathbf{C}_{\Theta<} \cup \mathbf{C}_{\Theta=} \cup \mathbf{C}_{\Theta>}$$

$$\text{及} \quad \mathbf{C}_{\bullet\Omega} = \mathbf{C}_{<\Omega} \cup \mathbf{C}_{=\Omega} \cup \mathbf{C}_{>\Omega}$$

就有下面的分解公式

$$\mathbf{C}(a) = \mathbf{C}_{.<} \cup \mathbf{C}_{.=} \cup \mathbf{C}_{.>} \text{ (参数形式)} \quad (5.1.6)$$

$$= \mathbf{N}_{.<} \cup \mathbf{N}_{.=} \cup \mathbf{N}_{.>} \text{ (主值形式)} \quad (5.1.7)$$

定义 5.1 *Problem XYZ* 是精确的, 如果对于它的每一个实例 *Instance XYZ*: $S = (P, r)$, 局部最优解 a 总是整体最优的。

把它写成一个特性如下 :

$$P_5^{(2)} : \text{Problem XYZ 是精确的。}$$

下面, 只考虑目标函数取得最大的情形。 其它情形也有类似的概念、定理和方法。

5.2 基本定理

把下面的三个不言自明的定理叫做基本定理。

定理 5.1 已知 $P_{12}^{(3)}$ -*Instance XYZ*: S 及 $f(U) = |U|$,

(i) 若 $(U, V) \in \mathbf{C}(a)$ 及 $b = a \Delta \tau(U, V)$, $b \in \mathbf{D}$ 。

(ii) 若 $(U, V) \in \mathbf{C}_{= \bullet}$, 则 $|a| = |b|$ 。

(iii) 若 $(U, V) \in \mathbf{C}_{> \bullet}$, 则 b 是 a 的一个参数得到改进的可行解。

定理 5.2 已知 $P_{12}^{(3)}$ -*Instance XYZ*: S ,

(i) 若 $(U, V) \in \mathbf{C}(a)$, 则 $b \in \mathbf{D}$ 。

(ii) 若 $(U, V) \in \mathbf{C}_{=}$, 则 $g(a) = g(b)$ 。

(iii) 若 $(U, V) \in \mathbf{C}_{>}$, 则 b 是 a 的主值得到改进的可行解。

定理 5.3 已知 $\{P_5^{(2)}, P_{12}^{(3)}\}$ -*Instance XYZ*: S 及 $f(U) = |U|$,

- (i) 可行解 a 的基数是最大的, 如果 $\mathbf{C}_{>\bullet} = \emptyset$ 。
- (ii) a 是唯一的基数最大的可行解, 当且仅当 $\mathbf{C}_{>\bullet} = \mathbf{C}_{=\bullet} = \emptyset$ 。
- (iii) 可行解 a 是最大的, 当且仅当 $\mathbf{C}_{\bullet>} = \emptyset$ 。
- (iv) a 是唯一的最大可行解, 当且仅当 $\mathbf{C}_{\bullet>} = \mathbf{C}_{\bullet=} = \emptyset$ 。

5.3 对称差分解(SDD)法

基于前面的讨论, 把下面的方法, 记作 $M_0^{(4)}$, 叫做对称差分解(symmetric difference decomposition method, SDD)法[15, 16, 18], 用来求解组合最优化问题。

1. 取一个初始可行解 a 。
2. 计算 $g(a)$ 。
3. 找出它的增量簇 $\mathbf{C}(a)$ 并作出它的分解公式。
4. 根据基本定理, 当前可行解 a 是最优的, 算出它的值, 停止。
5. 否则, 找出一个改进可行解, 把它作为当前解 a 。
6. 从第二步开始作下一轮的计算。

5.4 四个SDD法

把 $M_0^{(4)}$ 进一步具体化, 可以得到更明确的方法, 把它们记做 $M_j^{(4)} : \text{Method SDD-}j$ 。

第一个方法, $M_1^{(4)}$ 是

1. 取一个初始可行解 a 。
2. 计算 $g(a)$ 。
3. 从增量簇中 $\mathbf{C}(a)$ 逐一取出增量 (U, V) , 计算 $g(a\Delta\tau(U, V))$ 。
4. 如果 $g(a\Delta\tau(U, V)) \leq g(a)$, 从 $\mathbf{C}(a)$ 中删去 (U, V) ,
5. 否则, 把 $a\Delta\tau(U, V)$ 作为新的可行解。
6. 如果 $\mathbf{C}(a) = \emptyset$, 则当前可行解是最优的。

第二个方法, $M_2^{(4)}$ 是, 如果可行解 a 的值 $g(a)$ 满足 $P_{12}^{(3)}$, 且可以作 $\mathbf{C}(a) = \{(U^{(j)}, V^{(j)}) : 1 \leq j \leq t\}$ 。

1. 计算 $m = \max\{(g(V^{(j)}) - g(U^{(j)})) : 1 \leq j \leq t\}$ 。
2. 如果 $m \leq g(a)$, 停止。可行解 a 是最优的, 其值是 $g(a)$ 。

3. 否则,取一个 (U^*, V^*) 使得 $m = g(a\Delta\tau(U^*, V^*))$, 把 $a\Delta\tau(U^*, V^*)$ 作为新的可行解, 它的值等于 m 。

4. 从它作新一轮改进。

第三个方法是:

1. 任取 a 的一个简单自由碎片 V ,

2. 找出能够和它匹伍的所有简单可行碎片 $U^{(i)} (1 \leq i \leq s)$, 构成 a 的增量 $(U^{(i)}, V)$,

3. 作 $m = \max\{g(a\Delta\tau(U^{(i)}, V)) : 1 \leq i \leq s\}$ 。

4. 如果 $m > g(a)$,

5. 从 $(U^{(i)}, V)$ 中取一个 U^* , 使得 (U^*, V) 满足 $m = g(a\Delta\tau(U^*, V))$, 于是得到一个改进可行解 $a\Delta\tau(U^*, V)$ 。

6. 否则, 再取另一个简单自由碎片。

7. 通过所有简单自由碎片都不能得到改进可行解, a 就是最优解。

我们得到SDD法 $M_3^{(4)}$ 。

最后是:如果简单自由碎片都是单元素集合, 这就显得特别简单有效。 得到SDD方法 $M_4^{(4)}$ 。

5.5 几个例

例 5.1 最小支撑树问题[17] 设有一个典型实例 *Instance TREE* : $G = (V, E, r)$, 其中图 $G = (V, E, r)$ 是一个赋值连通图。 任一支撑树是可行解。其全体记作 D 。

设 $a, b \in D$, (a_b, b_a) 是互易对。 取 $e' \in b_a$, $e' \cup a$ 含有唯一的圈。 圈中除 e' 外诸边的长度最大者记作 e , 则 a 有一个增量 (e, e') 。 如果 $(e, e') \in \mathbf{C}_{\bullet <}$, 则 a 有一个改进可行解 $a\Delta\tau(e, e')$ 。 如果 $\mathbf{C}_{\bullet <} = \emptyset$, 边 e' 添加进 a 无法得到改进可行解, 则弃舍 e' 。 应用 $M_3^{(4)}$ 就得到调优算法。

例 5.2 二分图的(边数)最大匹配问题[17] 设有一个典型实例 *Instance MATCH*: $G = (V_1, V_2, E)$, 其中 $G = (V_1, V_2, E)$ 是一个二分图, 把匹配叫做可行解, 其全体记作 D 。

设 $a, b \in D$, (a_b, b_a) 是互易对。 图中每个顶点至多邻接 a 和 b 的边各一条。 导出子图 $a_b \cup b_a$ 中每个顶点的度数至多为2。 它的连通分支或是一个圈, 或是一条路(即人们熟知的交错路)。 如果 $|a_b| < |b_a|$, 就一定存在一条关于 a 的可扩路, 设它对应 (U, V) , 其中 $U \subseteq a_b$, $V \subseteq b_a$, 于是 $a\Delta\tau(U, V)$ 是一个改进可行解。 这就是Berg定理和匈牙利算法的出发点。

例 5.3 最短路问题, 平衡运输问题, 网络最大流问题, 最小费用最大流问题, 旅行商问题和中国邮路问题等等 也可作同样的讨论, 并利用SDD方法, 分别得到Bellman最优化原理, 三元运算(triple operation), Ford-Fulkson算法, 调优算法和管梅谷算法等等。

例 5.4 线性规划问题[16] 设有线性规划的典型实例 *Instance* LP: $S = (P, q, r)$ 。

它的任务是求解 $\max\{rx : Px = q, x \geq 0\}$

其中 P 是 $m \times n$ 矩阵, 它由列向量 $\{p_j : 1 \leq j \leq n\}$ 所组成, 仍记作 P 。 $q = [q_1, q_2, \dots, q_m]^T$ 是 m 维列向量, $r = [r_1, r_2, \dots, r_n]$ 是 n 维行向量。 r_j 是向量 p_j 的值。

令 $\alpha = \{\alpha_1, \alpha_2, \dots, \alpha_m\} \subseteq \{1, 2, \dots, n\}$, 相应地, $a = \{p_j : j \in \alpha\} \subset P$ 。由 a 的列向量组成的矩阵 A_α 如果有逆, a 是一个基。如果 $A_\alpha^{-1}q \geq 0$, a 是可行基, 它的值等于 $g(a) = r_\alpha A_\alpha^{-1}q$ 。我们知道, 如果实例有最优解, 它可以在一个可行基处达到。

设 P 是主集合, 满足条件 $A_\alpha^{-1}q \geq 0$ 的 $a \equiv a_\alpha \subset P$, $\alpha = \{\alpha_1, \alpha_2, \dots, \alpha_m\}$ (P 的一个子集合) 叫做可行解。实例的任务是在可行域 $\mathbf{D}$ 上求 $g(a) = r_\alpha A_\alpha^{-1}q$ 的最大值。因此线性规划是一个组合最优化问题。

设 $a, b \in \mathbf{D}$, 其中 $b \equiv b_\beta$, $\beta = \{\beta_1, \beta_2, \dots, \beta_n\}$ 。如果 $a \cap b = \{p_j : j \in \mu\}$, $\alpha = \lambda \cup \mu$, $\lambda \cap \mu = \emptyset$, 及 $\beta = \mu \cup \nu$, $\mu \cap \nu = \emptyset$ 。对于 b , 有 $A_\beta x_\beta = q$ 及 $g(b) = r_\beta A_\beta^{-1}q$ 。

定理 5.4 设 $\sigma_j \equiv r_j - r_\alpha A_\alpha^{-1}p_j$, 则有

$$g(b) - g(a) = \sum \{\sigma_j y_j : j \in \beta \setminus \alpha\} = \sigma_\nu y_\nu$$

给定可行解 $a \equiv a_\alpha$, 取自由向量 $p^{(j)} \in P \setminus a$ 。如果存在一个 $p_i^{(j)} \in a$ 使得 $(p_i^{(j)}, p^{(j)})$ 是一个增量, 于是 $g(a \Delta \tau(p_i^{(j)}, p^{(j)})) - g(a) = \sigma_s y_s$ 。所以有 $P_5^{(2)}$ 。

若 $\sigma_s y_s > 0$, 有 $(p_i^{(j)}, p^{(j)}) \in \mathbf{C}_{=>}$ 。 $a \Delta \tau(p_i^{(j)}, p^{(j)})$ 是 a 的一个改进。

可行解 a 是最大的, 若 $\mathbf{C}_{=>} = \emptyset$, 即, 不存在 $s \in \{1, 2, \dots, n\} \setminus \alpha$ 使得 $\sigma_s > 0$ 。这是人们熟知的改进单纯形算法, 也就是算法 $M_2^{(4)}$ 。这个结果似乎更令人有点意外。

6 求解初等组合最优化问题的方法的方法

至此, 我们从定义, 从三个最优化原理以及对称差分解法代数地得到了五类十六个方法。把它们列成一个树形结构是显然的。

我们建议: 给定一个问题实例, 按以下思路来研究。断定它确实是一个组合最优化问题的实例之后, 依次检查它

1. 是否满足第一最优化原理,
2. 是否满足第三最优化原理,
3. 是否能够从对称差分解法得到启发,
4. 能否建立同解法和隐枚举法,
5. 分支定界法是不得已才用的,

6. 枚举法只对规模很小的实例可用。

前四项中有一项是肯定的，就可以利用所属的方法发现相应的算法。再把所得到的那些算法进行计算复杂性的研究。如果前四项中一个也不满足，我们也许有更充分的理由怀疑问题不属多项式问题类。当然还有近似法，启发法，随机法等用来求解。

我们把上述建议叫做求解初等组合最优化问题的方法的方法。利用它来发现一个具体问题的实例的算法就成了这个实例的算法的算法。这是我们从[14]开始就梦寐以求的。

最后，作者感谢许多学者，特别费浦生，林诒勋，张福基，毛经中，唐国春诸位教授和周三明博士，感谢他们对作者的长期的帮助和鼓励，同时期盼读者们的批评和指正。

参考资料

- [1] Lawler E. L., Combinatorial Optimization: Networks and Matriods, Holt, Reinhart and Winston 1976.
- [2] 林诒勋，试论最优化原理的一般形式，曲阜师院学报(自然版) 3(1985)35.
- [3] 林诒勋(LinY.X.), An ordered independence system and its applications to Scheduling problems, European Journal of Operational Research, 74 (1994) 183-195.
- [4] 马仲蕃，条目“组合最优化“，《中国大百科全书 - 数学卷》第865页，1988.
- [5] 秦裕瑗，《Optimum Path Problems in Networks》，Hubei Education Press, 1992.
- [6] 秦裕瑗，Bellman 最优化原理—论动态规划 I，《应用数学》，7:3 (1994), 349-354.
- [7] 秦裕瑗，优化路问题的代数方法—论动态规划 II，《应用数学》，7:3 (1994), 410-416.
- [8] 秦裕瑗，决策论的两类判据，《管理工程学报》，8:3 (1994), 167-172.
- [9] 秦裕瑗，论优化问题的公理方法 I，《应用数学》，9:3 (1996), 261-265.
- [10] 秦裕瑗，论优化问题的公理方法 II – 算法原理与六个基本算法，《应用数学(增刊)》(1996), 9-12.
- [11] 秦裕瑗，论优化问题的公理方法 III – 多阶段决策问题，《数学杂志》，16:3(1996), 329-335.
- [12] 秦裕瑗，论优化问题的公理方法 IV – 有限改进算法与迭代算法，《数学杂志》17:3(1997), 325-330.
- [13] 秦裕瑗，论优化问题的公理方法 V – 优化集合的代数表达式，《数学杂志》17:3(1997), 331-334.
- [14] 秦裕瑗，算法的发现 (I) – 组合最优化的一个基本方法，《数学杂志》，14:3 (1994), 4336-444.
- [15] 秦裕瑗，算法的发现(II) – 对称差(的)分解法及其应用，《数学杂志》15:1 (1995), 82-88.
- [16] 秦裕瑗，郑肇葆，算法的发现(III) – 对称差(的)分解法的另一应用，《数学杂志》，18:1(1998), 76-80.
- [17] 秦裕瑗，郑肇葆，算法的发现(IV) – 论组合优化的特性清单，已被《数学杂志》，18:4(1998) 421-7.
- [18] 秦裕瑗，Symmetric-difference decomposition methods for combinatorial optimization, 《国际组合数学学术会议和夏季讲演会(合肥)》论文集1997.5.