中国科学(A 辑)
第24卷 第11期 SCIENCE IN CHINA (Series A) 1994年11月

一种求解合取范式可满足性问题的数学物理方法*

李 未 黄文奇
(北京航空航天大学计算机系，北京 100083) (华中理工大学计算机系，武汉 430074)

摘 要

给出合取范式与带电质点在某类静电场中势函数的一一对应关系，证明判断一个合取范式是否可满足等价于判断一个带电质点在相应的静电场中是否有使其势函数为零的位置，由于带电质点在静电场中总是沿使其势能下降最快的方向，也就是沿其势函数梯度指引的方向运动，并最终达到势能最低的位置．因此，对一个客观上可满足的合取范式所对应的势函数，使用梯度算法是求使该合取范式成真的赋值的高效算法．

关键词 合取范式、势函数、梯度算法

寻找求解 NP 完全问题的快速算法是当代理论计算机科学的核心问题之一．由于现代科技、军事以及工商管理的大量重要应用都归结为求解 NP 完全问题，因此，它又是计算机技术应用的瓶颈．合取范式的可满足性(简称 SAT)问题是一典型的 NP 完全问题．它的快速求解方法不仅具有重要的理论意义[1]，而且在软件自动开发技术、逻辑推理机、大规模集成电路的布线，以及知识库维护等许多领域都有直接的应用[2—4]．

人们研究合取范式的可满足性问题，是从寻找完全的解法开始的，也就是说这种算法能在有限步内确定一个合取范式是否是可满足的，如果可满足，则给出它的解．现在完全算法已有许多，穷举所有可能的赋值及 Davis-Putnam 算法是这方面的典型代表(文献[1, 5])．完全算法的缺点是它们的效率太低，基本上不能实用．数学与计算机科学近 20 年的研究表明，至少到现在，人们还未找到关于可满足问题的完全而又快速实用的求解算法[5]．在这种情况下，人们开始寻找不完全但却是快速而实用的算法[6—14]．这里所谓“不完全”的意思是指：如果范式是可满足的，则此算法将停机并给出一个解，否则算法可能不停机．

本文提出一种依据某类数学物理模型，寻找求解 NP 问题快速实用(但可能是不完全的)算法的方法，循此途径所得的算法称为 NP 问题的数学物理算法，又被称为拟物算法，因为这些算法是人们通过把待解决的问题与自然界或人类社会中的某些现象进行类比，找到待解决

1994-06-09 收稿，1994-08-14 收修改稿．
* 国家“八六三”高技术和国家自然科学基金资助项目．

问题的物理解释，又从这些现象的进化方式或人们处理这些现象的经验和理论中受到启发，产生某种求解策略，然后才制定出来的[7-11]. 更重要的是，当这种算法在某些特定场合计算效率有"疲软"表现时，人们还可以十分自然地利用这些类比和问题的物理意义，有针对性地对算法加以改进，从而得出更强有力的算法[11]. 事实上，递归论专著[15]中给出的许多策略都来源于人类的实践经验. 利用这些策略，数学家们对递归可枚举集作了深刻的研究，得出了大量的严格定理. 我们的研究实践也说明[7-11]，沿着这条途径，可以为许多 NP 完全问题或 NP 难度问题的求解设计出效率极高的数学物理算法.

本文将对合取范式的可满足性问题给出一种快速求解方法，并将着重描述是如何沿拟物途径，根据问题的物理数学意义，找出 SAT 问题的快速算法的，然后解释为什么它能得以快速求解的物理意义.

1　关于可满足性问题的算法

有关可满足性问题的算法研究是 80 年代末开始热起来的. Hooker 首先使用整数规划找到一些规模较小的可满足问题求解的算法[12]，Kamath 等人则提出了内点方法，并改进了 Hooker 的结果[13]. Gu 提出了离散的局部搜索方法[6]. 本节将给出两种最优化算法. 为此，首先叙述关于 SAT 问题的严格定义.

定义 1　合取范式及可满足性(SAT)问题

1. 给定一命题符集 $\mathscr{P}$，它由 m 个命题符 $P_1, \cdots, P_m$ 组成.

2. 给定一文字符集 $\mathscr{Q}$，一个文字符或是一命题符或是一命题符的非. 文字符用 Q 表示.

3. 给定一子句集 $\mathscr{C}$. 一个子句具有如下形式：

$$Q_1 \vee Q_2 \vee \cdots \vee Q_l,$$

子句用 Q 表示.

4. 合取范式集 $\mathscr{F}$ 一个合取范式(简称 CNF)具有如下形式：

$$C_1 \wedge C_2 \wedge \cdots C_n,$$

合取范式用 A 表示.

一个可满足问题是指：对给定的 CNF 是否存在一个关于命题符集的真值赋值使得 A 为真.

本文给出的近似算法的基本思路是由下述三步组成的.

1. 首先，把一个含有 m 个变元的合取范式(以下简称范式)看成定义域为$\{0, 1\}^m$，值域为$\{0, 1\}$的整函数 CNF_m，其中 0, 1 分别代表真假(或假真)值. 因此，这个范式的可满足问题就变为找一组变元的取值 $x \in \{0, 1\}^m$，使此 $\mathrm{CNF}_m(x)$ 达到极大(极小).

2. 这种看法初想起来对解决问题似乎没有多大帮助，因为求整函数的极值同样是很困难的问题. 如果再往前想一步，把这个关于 CNF_m 的整函数转换为一个实函数，称为目标函数，并且满足：此整函数达到最大(小)值的充分必要条件是(实)目标函数达到最大(小)值，则此合取范式的可满足问题就转换为求相应实函数的最大(小)值问题. 而对于实目标函数求最大(小)值问题，最优化理论和技术提供了大量的快速而有效的解决方法供选择和使用.

3. 众所周知，用最优化技术求最大(小)值的效率高低与目标函数直接有关. 也就是与把整函数 CNF_m 转换为什么样的实函数直接有关. 研究表明，目标函数能否选好的关键在于对

它的求最大(小)值问题是否有一好的物理解释．这也是为什么有些数学物理算法被称为拟物算法的原因．

本节将讨论两种转换方法，第一种是来自纯数学角度的转换，第二种则有其直接的物理意义．前者见文献[6]，后者则为本文提出．在下一节讨论近似算法和计算结果时将会发现，它们的计算效率有着数量级的差别．

变换 1.

这种变换的基本思路是将 $((P_1 \vee \neg P_2) \wedge (\neg P_1 \vee P_2 \vee P_3))$ 转换为 $x_1^2 \cdot (x_2-1)^2+(x_1-1)^2 \cdot x_2^2 \cdot x_3^2$. 用 T_1 表示这种变换，它可被递归地定义如下：

$$T_1: F_m \to R^m \to R$$

$$\begin{aligned}
T_1(C_1 \wedge \cdots \wedge C_n) &= T_1(C_1)+\cdots+T_1(C_n),\\
T_1(Q_1 \vee \cdots \vee Q_l) &= T_1^2(Q_1) \cdot \cdots \cdot T_1^2(Q_l),\\
T_1(P_i) &= x_i, \quad i=1, \cdots, m,\\
T_1(\neg P_i) &= x_i-1, \quad i=1, \cdots, m,\\
T_1(T) &= 0,\\
T_1(F) &= 1.
\end{aligned}$$

下面这个例子告诉我们变换 1 是如何工作的．

例 1

$$\begin{aligned}
&T_1((P_1 \vee \neg P_2) \wedge (\neg P_1 \vee P_2 \vee P_3))\\
&\quad = T_1(P_1 \vee \neg P_2)+T_1(\neg P_1 \vee P_2 \vee P_3)\\
&\quad = [T_1(P_1) \cdot T_1(\neg P_2)]^2+[T_1(\neg P_1) \cdot T_1(P_2) \cdot T_1(P_3)]^2\\
&\quad = x_1^2 \cdot (x_2-1)^2+(x_1-1)^2 \cdot x_2^2 \cdot x_3^2.
\end{aligned}$$

对变换1，用 $T_1(A)|_2$ 表示 $T_1(A)$ 在点 $(v(P_1),\cdots v(P_m))$ 的值，可以证明：

定理 1 令 A 为一 CNF，则赋值 v 使 A 满足的充分必要条件是 $T_1(A)|_2$ 达到最小值 0.

证 结论是显然的，由变换 T_1 产生的目标函数是非负函数，它的值标志 A 与"真"的偏离程度．

变换 T_2.

变换 T_2 的基本想法是将

$(p_1 \vee \neg p_2) \wedge (\neg p_1 \vee p_2 \vee p_3)$ 变成 $u_1(x_1, x_2)+u_2(x_1, x_2, x_3)$ 其中

$$u_1(x_1, x_2)=\begin{cases} (1-x_1)x_2, & \text{当 } x_1<1 \text{ 且 } x_2>0;\\ 0, & \text{否则};\end{cases}$$

$$u_2(x_1, x_2, x_3)=\begin{cases} x_1(1-x_2)(1-x_3), & \text{当 } x_1>0 \text{ 且 } x_2<1 \text{ 且 } x_3<1;\\ 0, & \text{否则}.\end{cases}$$

与变换 T_1 一样，T_2 也可以递归地定义如下：

$$T_2: F_m \to R^m \to R,$$

$$T_2(C_1 \wedge \cdots \wedge C_n)=T_2(C_1)+\cdots+T_2(C_n),$$

$$T_2(Q_1 \vee \cdots \vee Q_l)=\begin{cases} T_2(Q_1)\cdots T_2(Q_l), & \text{当 } T_2(Q_i)>0, \quad i=1, 2, \cdots, l;\\ 0, & \text{否则}.\end{cases}$$

$$
\begin{aligned}
T_2(P_i) &= 1-x_i, \quad i=1,\cdots,m,\\
T_2(\neg P_i) &= x_i,\\
T_1(T) &= 1,\\
T_1(F) &= 0.
\end{aligned}
$$

与变换 T_1 一样，变换 T_2 也有相应的定理.

定理 2　令 A 为一 CNF，则赋值 v 使 A 满足的充分必要条件是 $T_2(A)|_2$ 达到最小值 0.

证　不失一般性可以假设任意给定的有 m 个命题变元 $P_1,\cdots,P_m$ 有 n 个子句的合取范式可以写成

$$
\begin{aligned}
A &= C_1\wedge C_2\wedge\cdots\wedge C_n\\
&= (P_{1,1}\vee P_{1,2}\vee P_{1,k_1}\vee \overline{P}_{r1,1}\vee \overline{P}_{r1,2}\vee\cdots \overline{P}_{r1,k_{r1}})\wedge\cdots\wedge\\
&\qquad (P_{n,1}\vee P_{n,2}\vee\cdots P_{n,k_n}\vee \overline{P}_{rn,1}\vee \overline{P}_{rn,2}\vee \overline{P}_{rn,k_{rn}}).
\end{aligned}
$$

按变换 T_2 给出的相应的实目标函数为

$$U_2(x_1,\cdots,x_m)=\sum_{l=1}^{n} u_l(x_1,\cdots,x_m), \tag{1}$$

其中

$$
u_l(x_1,\cdots,x_m)=\begin{cases}(1-x_{l,1})\cdots(1-x_{l,k_l})x_{rl,1}\cdots x_{rl,k_{rl}},\\ \text{若 } x_{l,1}<1 \text{ 且}\cdots x_{l,k_l}<1 \text{ 且} x_{rl,1}>0 \text{ 且}\cdots x_{rl,k_{rl}}>0,\\ 0,\ \text{否则}.\end{cases} \tag{2}
$$

显然，若合取范式 A 可满足，则必有一真值指派使 CNF_m 的值为 1，由函数 $U_2(x_1,\cdots,x_m)$ 及变换 T_2 的定义，知函数 U_2 在该点的值必为 0. 反之，若目标函数 $U_2(x_1,\cdots,x_m)$ 的最小值为零，且某个最小值点为 $(x_1^*,\cdots,x_m^*)$. 对 $i=1,\cdots,m$，令

$$P_i^*=\begin{cases}1, & \text{若 } x_i^*\geqslant 1;\\ 0, & \text{若 } x_i^*<1\end{cases} \tag{3}$$

根据 $U_2(x_1,\cdots,x_m)$ 的定义，知它为一非负函数. 因此，对任意 l，$1\leqslant l\leqslant n$，有 $u_l(x_1^*,\cdots,x_m^*)=0$. 由定义至少有一 $x_{l,j}^*$ 满足 $1\leqslant j\leqslant k_l$，使 $x_{l,j}\geqslant 1$，或有一 $x_{rl,j}$ 满足 $1\leqslant j\leqslant k_{rl}$，使 $x_{rl,j}\leqslant 0$ 成立. 根据公式(3)，对前者有 $P_{l,j}^*$ 为真，对后者有 $P_{rl,j}^*$ 为假，从而 $\neg P_{rl,j}^*$ 为真. 总之，C_l 为真，$1\leqslant l\leqslant n$，因此，$(P_1^*,\cdots,P_m^*)$ 即为一个成真指派，故合取范式 A 可满足.

最后，需要指出：目标函数 U_2 不是从纯数学角度碰巧"猜"出来的. 它是从对一个物理现象，即带电质点在多维空间静电引力场中运动规律的类比中得到的；换句话说，目标函数 U_2 的最小值问题有一物理模型. 关于这一点，将在第四和第五节进行详细的论述.

2　梯度算法及计算结果分析

目标函数一旦选定，剩下的问题就是选择什么样的数值算法以便更快地求得最小值点的问题. 优化理论和技术为此提供了许多方法供选择. 本文所采用的是一种最基本的方法——梯度法. 这种算法的基本思路是用目标函数的梯度制导点 $(x_1,\cdots,x_m)$ 的走向，使目标函数的值迅速减小，并达到极小值. 这种基本想法可以用下述程序框架描述，其中 d 表示计算步长，U 代表目标函数，$v^{\ 1}*a$ 为 $U(x_1,\cdots,x_m)$ 在 v 的梯度的单位向量，ε 为步长的最小允许长度：

```
Grad-U
    {
     v=InitVect();
     step=0;
     d=sqrt(l)/100;
    loop
    {
      step ++;
      a=(-∂U/∂x_1, ..., -∂U/∂x_l);
      y=sqrt((∂U/∂x_1)^2+...+(∂U/∂x_m)^2)
      v'=v+y^{-1}*a*d;
      if (d≤ε ∧ (U(v')≮U(v))) break;
      if (U(v)>U(v')) d=d/2;
      v=v';
     }
     if (U(v)=0)return success; else return fail;
    }
```

令变换 T_1 和 T_2 产生的目标函数分别为 U_1 和 U_2，为了测试对目标函数使用梯度算法的计算效果，本文选用 Exact-3-SAT 模型，即合取范式的每个子句正好有 3 个文字，而子句中的每个文字符将从文字符集 Q 中以概率 $\frac{1}{2m}$ 选出．用随机产生的 3-SAT 合取范式做测试样本，是为了能客观地分析与比较各种算法的性能．

表 1 给出了 U_1 和 U_2 用梯度算法的比较结果．从表中可以看出：两个目标函数向最小值点逼近的时间复杂性均呈多项式型增长，而不是呈指数型增长；目标函数 U_2 比 U_1 的收敛速度要快一个数量级左右．我们将会看到其原因在于 U_2 向最小值的逼近有一合理的物理解释．U_2 的缺点是随着子句个数的增加，它的局部极值点也相应地增加．

本文还将基于实目标函数 U_2 的梯度算法与基于离散型目标函数的 local search 算法[6] 作了比较．基于 local seach 的算法也有不少，本文用的程序框架如下，它相当于文献[6] 中的 SAT1.3.

```
SAT1.3
    {
     v=InitVect();
     local=0;
     search=0;
     while (f(v)≠0 and local≤LIMIT)
```

```
{
 oldconflict = f(v);
 search ++;
 for (i=0; i≤m; i++)
 {
   v' = Swap(v_i, v̄_i);
   if (f(v) ≮ f(v)) v = v';
 }
 if (oldconflict = f(v)) or (search ≠ MAXSERACHSTEP and f(v) ≠ 0)
 {
  v = InitVect();
  local ++;
  search = 0;
 }
if (f(v) = 0) return success;
else return fail;
}
```

表 1　Grad-U_1 和 Grad-U_2 的比较结果[a)]

合取范式 (l, m, n)			执行时间(1/60s)						循环次数				样本数			
变元数	子句数	子句长度	最大执行时间		平均执行时间		最小执行时间		最大循环次数		平均循环次数		最小循环次数		成功次数	
			U_2	U_1	U_2	U_1	U_2	U_1	U_2	U_1	U_2	U_1	U_2	U_1	U_2	U_1
100	100	3	50	752	30	497	23	259	1	1	1	1	1	1	10	10
100	120	3	43	1210	32	719	27	355	1	1	1	1	1	1	10	10
100	140	3	51	1158	41	776	31	398	1	1	1	1	1	1	10	10
100	160	3	98	1588	62	953	48	475	1	1	1	1	1	1	10	10
100	180	3	77	1365	61	1051	46	761	1	1	1	1	1	1	10	10
100	200	3	105	1920	78	1384	60	921	1	1	1	1	1	1	10	10
100	220	3	1203	1812	204	1374	75	960	3	1	1	1	1	1	10	10
100	240	3	817	1568	268	1585	76	823	2	1	1	1	1	1	10	10
100	260	3	2917	2779	798	1989	111	1125	4	1	1	1	1	1	10	10
100	280	3	2794	3579	728	2078	126	970	4	1	1	1	1	1	10	10
100	300	3	10926	9241	2862	3757	153	1818	12	2	3	1	1	1	10	10
100	320	3	6472	10238	3935	7757	204	1492	8	2	4	1	1	1	10	10
100	340	3	12016	15596	4872	8797	231	1508	12	3	4	1	1	1	10	10
100	360	3	13626	56765	4335	10463	1649	1882	17	4	9	1	2	1	10	10
100	380	3	13586	19030	5560	14087	1534	2537	11	3	5	1	2	1	10	10
200	200	3	177	5061	129	2792	102	748	1	1	1	1	1	1	10	10
200	250	3	209	7164	164	4123	129	1865	1	1	1	1	1	1	10	10
200	300	3	1812	7024	422	4504	181	2864	2	1	1	1	1	1	10	10
200	350	3	2133	10512	503	5645	208	2964	2	1	1	1	1	1	10	10
200	400	3	2849	11650	1095	6258	351	3336	2	1	1	1	1	1	10	10
200	450	3	2632	12358	788	7504	342	3377	3	1	1	1	1	1	10	10
200	500	3	6772	11934	1972	9250	425	4951	3	1	1	1	1	1	10	10

a) 在 Sun Sparc 2 工作站上执行.

上述离散型 Local Search 方法中，目标函数 $f(v)$ 为赋值 v 使得 CNF 中子句为假的个数.

表 2 给出了 SAT1.3 与 Grand-U_2 的比较. 可以看出：当子句和变量个数较小时，SAT1.3 的执行时间要比 Grad-U_2 短；而当子句或变量个数增加时，SAT1.3 的平均执行时间呈非线性增长，其计算成功的可能性大大降低. 随着子句及变量个数增加，Grad-U_2 比 SAT1.3 的优越性也愈加明显，并显现出数量级的差别. 此外，SAT1.3 的最大执行时间与最小执行时间相差很大，而且与初始点的选择有关.

表 2 SAT1.3和 Grad-U_2 的比较结果[a)]

合取范式(l, m, n)			执行时间 (1/60 s)						循环次数						样本数			
变元数	子句数	子句长度	最大执行时间		平均执行时间		最小执行时间		最大循环次数		平均循环次数		最小循环次数		成功次数		超时次数	
			U_2	S1.3	U_2	S1.3	U_2	S1.3	U_2	S1.3	U_2	S1.3	U_2	S1.3	U_2	S1.3	U_2	S1.3
100	100	3	46	10	32	4	22	3	1	2	1	1	1	1	10	10	0	0
100	120	3	50	27	36	10	33	4	1	4	1	1	1	1	10	10	0	0
100	140	3	64	31	48	14	35	5	1	3	1	2	1	1	10	10	0	0
100	160	3	84	35	58	19	42	6	1	3	1	2	1	1	10	10	0	0
100	180	3	505	176	134	71	49	6	2	14	1	5	1	1	10	10	0	0
100	200	3	676	408	240	130	70	11	2	25	1	8	1	1	10	10	0	0
100	220	3	526	646	205	246	66	29	2	42	1	15	1	2	10	10	0	0
100	240	3	1177	2560	386	1041	62	13	3	139	1	53	1	1	10	10	0	0
100	260	3	5470	2621	849	824	78	82	5	130	2	39	1	4	10	10	0	0
100	280	3	8758	12257	1838	2478	131	17	6	491	2	101	1	1	10	10	0	0
100	300	3	1473	25603	416	6965	140	367	2	877	1	239	1	13	10	10	0	0
100	320	3	10221	186078	3602	61880	119	387	10	5899	4	1999	1	13	10	10	0	0
100	340	3	7862	19195	3808	9532	168	4419	9	622	3	313	1	145	8	4	2	6
100	360	3	15529	26105	8851	13110	1255	274	10	783	5	396	2	9	8	4	2	6
100	400	3	16586	25218	7270	23387	1430	21557	10	653	4	606	1	559	4	2	6	8
100	420	3	16714	∞	9504	∞	255	∞	12	∞	6	∞	1	∞	3	0	7	10
100	440	3	16638	40969	16565	40969	16492	40969	12	964	11	964	10	964	2	1	8	9
200	200	3	267	127	141	54	77	19	1	3	1	1	1	1	10	10	0	0
200	240	3	243	580	161	105	94	21	1	10	1	2	1	1	10	10	0	0
200	280	3	3706	580	633	209	135	23	2	9	1	3	1	1	10	10	0	0
200	320	3	4844	6573	704	1155	183	32	2	78	1	14	1	1	10	10	0	0
200	360	3	2393	5124	708	1712	235	147	2	52	1	18	1	2	10	10	0	0
200	400	3	3450	22517	721	4658	341	60	2	199	1	41	1	1	10	10	0	0
200	440	3	4911	112671	900	23713	307	3291	3	856	1	185	1	26	10	10	0	0
200	480	3	7736	402015	1597	102744	307	9751	5	2665	2	701	1	69	10	10	0	0

a) 在 A Sun Sparc 2 工作站上执行.

需要指出的是：本文省略了梯度算法与 Davis-Putnam 算法的比较，因为当子句或变量个数增加到一定数量时，Davis-Putnam 算法的执行时间要远比 Grad-U_2 算法长得多.

3 CNF 可满足问题与覆盖问题[10]

本节将通过一个简单的例子给出 CNF 的可满足问题与覆盖问题[10] 的关系. 考虑下述合取范式：

$$(P_1 \vee P_2) \wedge (\neg P_1 \vee P_2) \wedge (\neg P_1 \vee \neg P_2). \tag{4}$$

与第二节一样，逻辑“真”与“假”分别用自然数“1”及“0”标志．考虑二维实平面到 $P_i, i=1, 2$ 赋值的变换：

$$P_i=\begin{cases}1, & 若\ x_i\geqslant 1;\\ 0, & 若\ x_i\leqslant 0;\\ 随意给定\ 0\ 或\ 1, & 若\ 0<x_i<1.\end{cases} \tag{5}$$

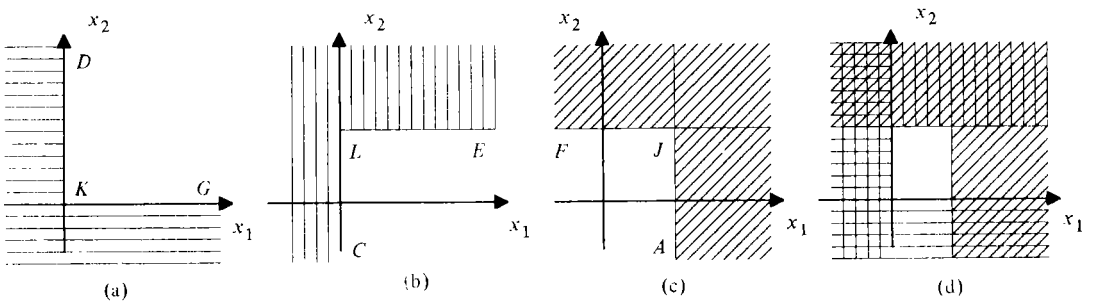

图 1

(a) $x_1\leqslant 0,\ x_2\leqslant 0$.　(b) $x_1\leqslant 0,\ x_2\geqslant 1$.　(c) $x_1\geqslant 1,\ x_2\geqslant 1$　(d) $0\leqslant x_p x_2\leqslant 1$

在此转换之下，显然，合取范式 4 中第三个子句 $\neg P_1\vee\neg P_2$ 成真的充要条件为 $x_1\leqslant 0$ 或 $x_2\leqslant 0$，或者说点 (x_1, x_2) 处于平面的第二、三、四象限，即游动点 (x_1, x_2) 被图 1(a) 中有水平阴影线的以 GKD 为边界的固定闭区域 III 所覆盖．同理，合取范式 4 中第二个子句 $\neg P_1\vee P_2$ 成真的充要条件为 $x_1\leqslant 0$ 或 $x_2\geqslant 1$，即游动点 (x_1, x_2) 被图 1(b) 中带有垂直阴影线的以 ELC 为边界的固定闭区域 II 所覆盖；合取范式 4 中第一个子句 $P_1\vee P_2$ 成真的充要条件为 $x_1\geqslant 1$ 或 $x_2\geqslant 1$，即游动点 (x_1, x_2) 被图 1(c) 中带有斜阴影线的以 AJF 为边界的固定区域 I 所覆盖．因此，为合取范式 4 找一个成真赋值的问题等价地变为在实二维欧氏空间中找一个点 (x_1, x_2) 使其被三个已知区域 I, II, III 所共同覆盖的问题．

4　目标函数 U_2 的物理意义

设想上节提到的区域 I、II、III 分别为带有正电荷的金属板，金属板与金属板之间用绝缘层隔开．金属板为导体，其中的静电势皆相同，设带有单位负电荷的质点 P 在平面上的坐标为 (x_1, x_2)，它受到分别来自三个金属板 I、II、III 的三个静电引力 $\boldsymbol{F}_1$, $\boldsymbol{F}_2$, $\boldsymbol{F}_3$ 的作用．质点 P 在这三个力的驱使下运动．每一个金属板都有一将质点 P 吸引到处于本板之上的要求，只要质点 P 不在此板上，它就受到这块板引力的作用；直到质点处于本板之上，此金属板对质点的引力才消失．这时质点又会受到其他金属板的引力作用，向这些金属板运动．总之，一旦某块金属板的要求已被满足，则它对质点的吸引立即停止，并将机会让给别的金属板．这种基于静电引力与金属屏蔽现象的运动图案，恰好与现代递归论中逐步满足某种相互制约与关联要求的“有穷损害优先策略”(finite injury strategy) 的精神是一致的[15]．

这种运动图案可以精确地描述如下：设三个带电金属板 I, II, III 分别产生自己的静电势场为 $u_1(x_1, x_2)$, $u_2(x_1, x_2)$ 及 $u_3(x_1, x_2)$．因此，系统的总静电势场 $\nabla(x_1, x_2)$ 为

$$\nabla(x_1, x_2)=u_1(x_1, x_2)+u_2(x_1, x_2)+u_3(x_1, x_2). \tag{6}$$

带电质点 K 受到的静电引力为

$$\boldsymbol{F}=\boldsymbol{F}_1+\boldsymbol{F}_2+\boldsymbol{F}_3=-\mathrm{grad}\nabla=(-\mathrm{grad}u_1,\ -\mathrm{grad}u_2,\ -\mathrm{grad}u_3) \tag{7}$$

其中 grad 为梯度算符，∇ 的梯度向量为

$$\mathrm{grad}\nabla(x_1,x_2)=\left(\frac{\partial\nabla(x_1,x_2)}{\partial x_1},\ \frac{\partial\nabla(x_1,x_2)}{\partial x_2}\right). \tag{8}$$

现在我们具体推导出 (6) 式中 $u_3(x_1,x_2)$ 的解析表达式．由静电学理论知，$u_3(x_1,x_2)$ 在第二、三、四象限，即在金属内部，取值为零；$u_3(x_1,x_2)$ 在第一象限即是在真空中，满足 Laplace 方程，于是对 $u_3(x_1,x_2)$ 的定解条件为

$$\begin{cases}\dfrac{\partial^2u_3(x_1,x_2)}{\partial x_1^2}+\dfrac{\partial^2u_3(x_1,x_2)}{\partial x_2^2}=0,\ \text{当点}(x_1,x_2)\text{在第一象限内部；}\\ u_3(x_1,x_2)=0,\ \text{当点}(x_1,x_2)\text{在第一象限的边界线上．}\end{cases} \tag{9}$$

现按分离变量法求非零解．令 $u_3(x_1,x_2)=v_1(x_1)v_2(x_2)$，代入 (9) 式得

$$v_1''(x_1)v_2(x_2)+v_1v_2''(x_2)=0 \text{ 即 } v_1''(x_1)/v_1(x_1)=-v_2''(x_2)/v_2(x_2).$$

于是

$$v_1''(x_1)/v_1(x_1)=-v_2''(x_2)/v_2(x_2)=C,$$

此处 C 为常数．令 $C=0$，得 $v_1''(x_1)=0$，$v_2''(x_2)=0$．于是 $v_1(x_1)=c_{11}x_1$，$v_2(x_2)=c_{21}x+c_{22}$，得

$$u_3(x_1,x_2)=(c_{11}x_1+c_{12})(c_{21}x_2+c_{22}).$$

由边界条件及 $u_3(x_1,x_2)$ 为非零解，定出 $c_{12}=c_{22}=0$，于是 $u_3(x_1,x_2)=c_{11}c_{21}x_1x_2$，知函数 $u_3(x_1,x_2)=x_1x_2$ 为满足定解条件 (9) 的非零解．从而静电势函数 $u_3(x_1,x_2)$ 在全平面上的解析表达式为

$$u_3(x_1,x_2)=\begin{cases}x_1x_2, & \text{当 } x_1>0 \text{ 并且 } x_2>0;\\ 0, & \text{当 } x_1\leqslant 0 \text{ 或者 } x_2\leqslant 0.\end{cases} \tag{10}$$

$u_3(x_1,x_2)$ 恒非负，$u_3(x_1,x_2)$ 的值代表金属板 III 对坐标为 (x_1,x_2) 的质点 P 静电引力势．同理得出静电势函数 $u_2(x_1,x_2)$ 在全平面的解析表达式为

$$u_2(x_1,x_2)=\begin{cases}x_1(1-x_2), & \text{当 } x_1>0 \text{ 并且 } x_2<1;\\ 0, & \text{当 } x_1\leqslant 0 \text{ 或者 } x_2\geqslant 1.\end{cases} \tag{11}$$

静电势函数 $u_1(x_1,x_2)$ 在全平面的解析表达式为

$$u_1(x_1,x_2)=\begin{cases}(1-x_1)(1-x_2), & \text{当 } x_1<1 \text{ 并且 } x_2<1;\\ 0, & \text{当 } x_1\geqslant 1 \text{ 或者 } x_2\geqslant 1.\end{cases} \tag{12}$$

(10)—(12) 式的叠加即是体系的总势能 $\nabla(x_1,x_2)$ 的解析表达式．

显然，本节的 u_1,u_2,u_3 及 $\nabla(x_1,x_2)$ 与上节对区域 I, II, III 的描述、第二节的目标函数 U_2 完全一致．这也就是说，如果在 (x_1,x_2)，质点的势函数为零，即 $\nabla(x_1,x_2)=0$，则此点被三个区域同时覆盖，也就是与此点对应的真值赋值，使与 $\nabla(x_1,x_2)$ 对应的合取范式为真．

由于带电质点在静电场中总是沿使其势能下降最快的方向，也就是沿其势函数梯度指引的方向运动，并最终达到某个势能“局部”最低的位置．因此，对目标(势)函数 U_2，使用梯度算法总能以最快的速度达到某个极小值．关键问题是只有当质点到达使势函数为 0 的位置，其相应的合取范式才为真．为此，当带电质点达到某个势能“局部”最低的位置时，必须给它一点能量，使之“跳出”势能“局部”最低位置，然后再沿梯度方向重新寻找势能“绝对”最低位置．

但即使质点到达了势能"绝对"最低的位置，也不等于说在此处势函数为 0. 事实上，势函数最低值是否为 0，不是由梯度算法决定的，而是由金属板的分布决定的，或者说由合取范式客观的可满足性决定的. 梯度法在这种意义上是不完全的算法.

以上是标法 Grand-U_2 的物理意义. 据知，这是当前国际上已公开发表的合取范式可满足性求解的效率最高的优化算法. 最后，需要指出的是本文在方法论上的意义，这就是：寻找有物理解释的优化算法，是求解 NP 问题的重要途径，它有时比没有客观对象的纯粹数学猜想为好.

致谢　作者感谢吕卫峰为比较本文涉及到的三种算法进行了大量的计算工作，并制成本文中的两张表格.

参　考　文　献

[1] Garey, M. R., Johnson, D. S., *Computers and Interactability: A Guide to the Theory of NP Completeness*, Freeman, San Francisco, 1978.
[2] Hochbaun, D. S., Maass, W., *J. of the Association for Computing Machinery*, 1985, **32**(1): 130—136.
[3] 李　未，中国科学，A 辑，1992, (10): 1103—1114.
[4] Li, W., *Programming Languages and Systems*, ESOP'94, LNCS 788, Springer-Verlag, 1994, 394—409.
[5] Dunham, B., Wang Hao, *Annals of Mathematical Logic* 1976, 117—154.
[6] Gu, J., *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 23, No. 4, July/August, 1993.
[7] 黄文奇、詹叔浩、应用数学学报，1979, (2): 176—180.
[8] 黄文奇、陈　亮，中国科学，A 辑，1991, (3): 325—331.
[9] Huang Wen-Qi, Zhan Shu-Hao, *Mathematical Reviews*, 1982, **82h**: 52002.
[10] 黄文奇，计算机学报，1989, (9): 610—680.
[11] 黄文奇、朱　虹、许向阳等，计算机学报，1993, (11): 829—836.
[12] Hooker, J. N., Fedjki, C., *Tech. Rep.* 77-88-89, GSIA, Carnegie Mellon Univ, Aug., 1989.
[13] Kamath, A. P. *et al.*, *Mathematical Science Research Center*, AT&T Bell Labs, Oct., 1989.
[14] Rogres, C. A., *Packing and Covering*, Cambridge University Press, Cambridge, 1964.
[15] Soare, R. I., *Recursively Enumerable Sets and Degrees*, Springer-Verlag, Berlin, 1987.