南京大学数学系 Department of Mathematics Nanjing University
何 炳 生 教授 Prof. Dr. Bingsheng HE
邮编：210093 Email:hebma@nju.edu.cn Homepage: http://maths.nju.edu.cn/~hebma

在研究乘子交替方向法（ADMM）之前，我的主要工作是求解单调变分不等式的投影收缩算法。交通网络分析中的数学问题往往归结为变分不等式，香港科技大学的杨海教授从我发表的论文知道我研究变分不等式的求解方法，邀请我 1997 年去香港科大访问。阅读了一些交通研究的文献以后，我们开始了 ADMM 求解变分不等式的研究。近 10 年来，ADMM 被广泛应用于求解带可分离结构的凸优化问题，成了热门课题，也激发了我们进一步的研究兴趣。

我的第一篇关于交替方向法的文章[1] 1998 年在 OR Letter 上发表，是跟杨海教授合作的。这篇文章主要讨论乘子交替方向法中，增广拉格朗日函数中罚参数的选取。人们往往只考虑固定的参数，我们证明了在罚参数单调不减或者单调不增下有界的情况下的收敛性。[2] 给出了自适应选取罚因子的一个准则，文章 2000 年发表在 JOTA 上。这个“源于挑担两头要差不多”的简单准则，被 Stanford 大学 S. Boyd 教授 (Boyd 教授是美国工程院院士，2006 年世界数学家大会邀请报告人，2017 年当选为中国工程院外籍院士) 在 2010 年的一篇综述文章中称为一个简单而有效的公式 (A simple scheme that often works well)，对我们的分析依据也作了简要介绍。他们近年开发的凸优化求解器 SnapVX 的说明文章中也注明参考了我们的调比法则。

计算数学的，往往对问题不精确求解更感兴趣，有关论文[3] 我们 2002 年发表在 MP 上。在交替方向法求解过程中加了正则项，子问题强单调所以允许不精确求解，理论上正则项还能变动，只是其中的误差平方可加是为了满足审稿人的要求才那样做的，作为计算数学工作者，我更重视的是相对误差。论文[4]通过对一些具体问题的讨论，说明带权的罚参数选择是必须的。论文[5]给出的 ADMM 类预测-校正方法，其中的预测通过简单投影实现的，采用了相对误差的策略。预测-校正是我们从研究投影收缩算法开始的一贯的思想。

虽然凸优化的最优性条件就是一个单调变分不等式，2008 年以前，我的着眼点还是求解管理科学中不能轻易转换成优化问题的一些变分不等式，总觉得优化问题应该有自己更简单的方法。

2008 年以后，ADMM 被广泛应用于求解带可分离结构的凸优化问题，成了热门课题，也激发了我们进一步的研究兴趣。理论上，我们分别证明了乘子交替方向法遍历意义下[6]和点列意义下的 O(1/t) 收敛速率[9]，并且将这些性质推广到一般的 Douglas-Rachford 算子分裂法上[10]。文章分别发表在 SIAM Numerical Analysis, Numerische Mathematik 和 MP 上。证明都相当简单，三篇文章加起来不到 30 页。其中发表在 SIAM Numer. Anal.上的文章，发表后至今都在该刊的 20 篇 Most frequently read papers 中，常常在这个数据库中排名第二至第六。

在方法上，我们首先考虑了交换 y 子问题求解和乘子校正次序的交替方向法 [7]，发表在中国科学。这样的交替方向法可以解释成 PPA 算法而延拓，很多情况下可以提高计算效率。

由于乘子交替方向法中的 x 和 y 子问题从本质上讲是平等的，我们给出了做完一个子问题就校正一次拉格朗日乘子的“对称的乘子交替方向法”， 2014 年发表在 SIAM Optimization [8]，论文是和袁晓明及 Princeton 大学的两位学者合作的，年轻人做了大量的计算工作。这篇论文的三、四节的理论证明，后来被总结成凸优化分裂收缩算法的统一框架，我主页上的报告三和综述文章《我和交替方向法 20 年》中都有介绍。

论文 [11] 关于“Generalized ADMM”，同样是跟袁晓明和 Princeton 大学的两位年轻学者合作的，年轻人做了大量的计算工作，2015 年发表在 Mathematical Programming Computation 上。

论文 [12] 对“Symmetric version of ADMM”做了进一步的研究，对两次校正步长的范围做了进一步的研究，2016 发表在 SIAM Imaging Science 上。

对交替方向法的起源、跟增广拉格朗日乘子法的关系，以及收敛性证明感兴趣的读者，请读我的二十集讲义中的第 11 讲，那里的证明相当简单。对收敛速率感兴趣的读者阅读我们的论文[6]和[10]；对算法发展感兴趣的建议阅读论文 [7] 和 [8]。