行星际脉冲转移轨道设计与优化算法

戴光明 彭 雷 罗治情 著

XINGXINGJI
MAICHONG ZHUANYI GUIDAO SHEJI
YU YOUHUA SUANFA

中国地质大学出版社有限责任公司
ZHONGGUO DIZHI DAXUE CHUBANSHE YOUXIAN ZEREN GONGSI

行星际脉冲转移轨道设计与优化算法

戴光明　彭雷　罗治情 著

中国地質大學出版社有限责任公司
ZHONGGUO DIZHI DAXUE CHUBANSHE YOUXIAN ZEREN GONGSI

图书在版编目(CIP)数据

行星际脉冲转移轨道设计与优化算法/戴光明,彭雷,罗治情著.—武汉:中国地质大学出版社有限责任公司,2012.4

ISBN 978-7-5625-2804-3

Ⅰ.①行…

Ⅱ.①戴…②彭…③罗…

Ⅲ.①行星际轨道-设计②行星际轨道-最优化算法

Ⅳ.V412.4

中国版本图书馆 CIP 数据核字(2012)第 062091 号

行星际脉冲转移轨道设计与优化算法 戴光明 彭雷 罗治情 著

责任编辑:段连秀 策划编辑:段连秀 责任校对:张咏梅

出版发行:中国地质大学出版社有限责任公司(武汉市洪山区鲁磨路 388 号) 邮政编码:430074

电 话:(027)67883511 传 真:67883580 E-mail:cbb@cug.edu.cn

经 销:全国新华书店 http://www.cugp.cug.edu.cn

开本:787 毫米×980 毫米 1/16 字数:240 千字 印张:9

版次:2012 年 4 月第 1 版 印次:2012 年 4 月第 1 次印刷

印刷:武汉教文印刷厂 印数:1—1 000 册

ISBN 978-7-5625-2804-3 定价:25.00 元

前 言

深空探测已成为世界关注的焦点，这类任务的高成本特性，也使得与探测成本密切相关的探测轨道的设计和优化方法的研究成为热点。深空探测的轨道（尤其在涉及借力飞行变轨时）通常有多种转移方案，这使得优化空间呈现急速增长的现象。轨道优化算法经历了由传统的确定性算法到智能优化算法的发展过程。

轨道优化本质上就是求解一类带约束的非线性函数优化问题。早期的求解方法包括离散化的方法、配点法、打靶法、单纯形法，还有根据梯度计算的方法，如序列二次规划方法、共轭梯度法等算法。但是轨道优化的目标函数往往具有类周期特性，且大部分局部最优解附近都有若干伴生解存在，使得这些确定性算法在应用过程中遇到了病态梯度、初始点敏感、局部收敛等问题，难以满足实践的需要，而寻求一种全局优化算法变得越来越重要。

演化算法是一种新兴的优化工具，它是借鉴自然界中进化与遗传机制的一种优化算法，主要用于解决复杂的工程技术问题。具有自适应搜索、渐进式搜索及并行式搜索的特点，并且有很强的鲁棒性，是一种全局的智能搜索方法。在轨道设计领域，越来越多的问题也开始采用演化算法或其衍生方法求解，美国国家航空和宇宙航行局（National Aeronautics and Space Administration，NASA）和欧洲航天局（European Space Agency，ESA）还有专门机构设置相关课题开展此类问题研究，取得了很多创新性的研究成果。

为了总结该领域的研究成果，推动国内在该领域的研究与应用，我们编著本书，以期抛砖引玉。本书将结合我们科研团队的研究成果，从行星际轨道动力学基础、行星际脉冲转移轨道设计、行星际脉冲转移轨道优化模型、优化算法和仿真系统等方面进行较为系统全面的阐述和讨论。

本书共分九章。

第一章讨论了轨道设计的基本概念，结合轨道设计优化问题对最优化算法进行了总结。

第二章对轨道计算的背景和基础知识进行了介绍，包括常用时空系统、航天器动力学原理以及机动和变轨，为描述后续研究工作做了铺垫。

第三章分析了行星际脉冲转移轨道设计模型，详细介绍了基于二体轨道设计的圆锥曲线拼接法具体计算方法。圆锥曲线拼接法将轨道分成日心段和行星质心段进行分段计算，分段计算所涉及到的轨道衔接问题（如不同惯性坐标系之间的转换）是圆锥曲线拼接法需要解决的关键问题之一。在深空机动（Deep Space Maneuver，DSM）的介绍中，利用位置模型 DSM，来参数化非共面 Lambert 问题中的大倾角转移轨道面临的高能量消耗问题。借力飞行轨道在行星际轨道设计中是极有价值的一种轨道设计方案，利用借力星体引力的甩摆效应可以减少对能量的需求。本书引入常用于精确轨道确定中的 B 平面方法进行无机动借力飞行轨道面和航天器接近目标行星的轨道面确定。打靶法是解决多体积分的一种数学方法。当目标点接近目标星体时，为了获得较高的打靶精度，我们提出了一种逐次逼近打靶法。

第四章讨论了如何将轨道设计方案参数化为优化算法评估函数的决策向量，分析了 MPGA（Multiple Powered Gravity Assist）＋位置模型 DSM 和 MGA（Multiple Gravity Assist）＋速度模型 DSM 两种多借力飞行行星际脉冲轨道优化模型，并分析了这些模型的时间复杂度。如何对转移方案进行参数化，并评价方案的优劣成了轨道优化的首要问题。

第五章给出了使用差分演化（Differential Evolution，DE）算法进行的大量实验，验证了双脉冲轨道优化问题基础上加入 DSM（位置模型或速度模型）可有效降低任务的燃料消耗。

第六章和第七章分别介绍了序列二次规划（Sequential Quadratic Programming，SQP）方法和单纯形算法（Simplex Method）在行星际脉冲转移轨道优化设计中应用。SQP 方法是一种常用的求解单目标非线性带约束规划问题的方法，求解速度快，运算精度高，是求解无约束最优化问题的牛顿法和拟牛顿法对约束优化问题的推广，是一种局部的精确搜索方法。单纯形算法是求解优化问题

的直接搜索寻优方法。针对这两种经典确定性算法的局部搜索特征，将其与演化算法相结合，使其全局搜索能力得到了提高。

第八章给出了 NSGA－Ⅱ算法对双脉冲变轨问题进行的多目标优化实验结果。燃料消耗和飞行时长通常是行星际轨道优化算法需要同时考虑的目标。

第九章讨论了我们自主研发的，具有自主知识产权的行星际轨道优化仿真工具软件，主要从软件的结构、数值仿真、可视化仿真、任务管理和任务优化等方面进行了介绍。

本项研究工作得到了“十一五”、“十二五”国防科技工业民用专项科研技术研究项目等的资助，在此深表谢意！

中国国际战略学会战略研究中心陶家渠研究员为本书的研究工作提供了很多的指导和帮助。中国地质大学（武汉）空间任务分析与设计科研团队李晖、王茂才、胡霍真、武云、石再明、王于、李艳、王剑文、陈方杰、周蹭、谭毅等做了大量细致的研究工作，专著中饱含有他们的智慧和成果。

由于笔者水平有限，书中不足和错误不可避免，恳请专家和各位读者批评指正。

作 者

2011 年 8 月

目 录

第一章　概　论

§1.1　概　述

随着科学技术的进步与经济的发展，全球范围的航天活动正在蓬勃发展。根据航天任务的不同，人类航天活动主要分为三大重点发展领域：地球应用卫星、载人航天和深空探测。深空探测通常是为了获得翔实可靠的科学数据，利用深空探测器脱离地球引力场，进入太阳系和宇宙空间，通过接近目标的方式，对地球以外的其他天体（如行星、彗星和小行星等）开展探测活动。其五个重点领域是：月球探测、火星探测、水星与金星探测、巨大行星及其卫星探测和小行星与彗星的探测（马文臻，2006）。

通过深空探测，能帮助人类研究太阳系及宇宙的起源、演变和现状，进一步认识地球环境的形成和演变，认识空间现象和地球自然系统之间的关系，为人类提供保护地球和进入宇宙的可能。从现实和长远来看，对深空的探测和开发具有十分重要的科学和经济意义。深空探测技术作为人类进行空间资源开发与利用、空间科学与技术创新的重要途径，成为一个国家综合国力和科学技术发展水平的重要特征与标志，已引起世界各国的极大关注。

深空探测已经成为当今世界各国科学研究的热点。由于航天技术关系到一个国家的经济发展和国家战略安全，关系到一个国家未来的发展空间，对世界政治、经济、军事及科学技术均具有重大的影响。各国都在调整发展战略，加大投入，加速发展航天技术。尤其是NASA 和 ESA，为了在空间研究中保持自己的主动地位，不断调整自己的研究目标，展开了空间科学研究领域的竞争。

为了适应国际航天领域发展趋势，推动我国航天技术和航天事业的发展，对和平利用外层空间和人类的文明和进步做出应有的贡献，根据科学研究和提高综合国力的现实需求，我国明确提出了“发展空间科学，开展深空探测”的发展目标，包括建立新型的科学探测与技术实验卫星系列，加强空间微重力、空间材料科学、空间生命科学、空间环境和空间天文研究，以及开展以月球探测为主的深空探测的预先研究。近期月球探测正在实施当中，月球探测将成为我国空间科学和技术发展的第三个里程碑。发射人造地球卫星、载人航天和深空探测是我国航天活动的三部曲，以后将会陆续开展行星际的深空探测和研究。

深空探测任务的首要关键技术之一是航天器轨道设计。飞行轨道可以是直接转移轨

道、持续小推力转移轨道(如太阳光帆和离子推进器等)、通过绕飞行星进行借力飞行(Swing - by 或 Gravity Assist,GA)变轨的轨道,或利用引力平衡点进行转移的节能转移轨道(刘林等,2005)。但最终目的都是为了设计出飞行时间尽量短,飞行所需燃料尽量少的飞行轨道。

深空探测任务中,航天器摆脱地球引力,进入日心转移轨道,最后与目标行星相遇,这种轨道过渡顺序称为行星际过渡。行星际探测中,航天器需要经历脱离地球和接近目标行星的过程。这些过程中,受到引力主要来自行星(远大于太阳引力对其的影响),人们通常将行星引力占主导作用的这部分范围称作引力影响球(耿长福,2006)。因此深空探测任务轨道通常也分为三段:逃逸轨道、日心过渡轨道和目标行星俘获段(杨嘉墀等,2005),若涉及借力飞行变轨时,还需要增加一段,即行星绕飞段。

由于深空探测任务的目标星体距离地球很远,如果采用相对目标星的直接转移轨道,那么所要求的发射能量会很大,目前的运载火箭可能无法满足其要求。为了节省发射能量,通常先用较小的速度飞行,然后在飞行过程中借助行星的引力来加速或改变航天器飞行方向,从而最终飞向目标星,这种借助行星引力而实现变轨飞行,被称为借力飞行。这就是说,在行星际中可以利用行星的引力作用改变航天器的运动速度的大小和方向,从而可以在没有任何动力消耗的情况下对航天器加速,从而实现变轨。

“先驱者”、“旅行者”和“伽利略”探测器的轨道已经证明了借力飞行轨道在行星际任务中是极有价值的一种轨道设计方案,借力飞行技术目前已应用于行星际的轨道设计中。

国际上多天体交会借力飞行轨道设计的主要方法通常是:通过绘制发射和到达能量等高线图(Sims, *et al*,1996)来寻找满足发射要求的发射窗口,通过在借力天体处飞入和飞出能量的匹配将不同的轨道段用圆锥曲线法拼接起来,然后根据飞行时间或发射能量等约束进行优化设计。Brouke 曾对借力飞行技术做了详细的描述,Sergevevsky 和 Yin 等人提出了绘制确定直接转移轨道的发射和到达能量的 Pork Chop 图的方法,后人沿用此法并不断改进(马文臻,2006)。

能量等高线法原理简单直观,并能对发射机会做出一个全局的估计,应用十分广泛。NASA 甚至将部分时段向各大行星飞行的能量等高线图绘制成手册供轨道设计者使用。但是能量等高线法本质上是一种穷举的方法,需要耗费大量计算时间,并且无法精确控制计算的精度,若要提高精度就需要将时间节点划分得更密,计算量也会随之成倍增长(石再明等,2006)。因此迫切需要开发一类高效的算法,能够在大的优化空间中快速搜索到全局最优解(任远等,2007)。目前,从轨道设计模型到优化算法,从学术界到工程领域都有许多研究成果,其中大量涉及轨道交会问题的研究。近地航天器的交会与深空探测交会模型有所不同,如较近的航天器使用基于相对运动的 CW 方程(又称 Hill 方程)进行求解(向开恒等,1999;朱仁璋等,2005;吴涛等,2001),其中也有基于绝对运动方法的求解模型,朱仁璋等(2006)对航天器交会的不同模型进行了比较性研究。

对新型借力飞行模型和基于该模型的轨道优化设计工作，国内外也有相关研究，如利用行星大气进行气动借力飞行的行星任务设计，哈尔滨工业大学的乔栋等(2005b)、国外的Povoleril等(2004)都开展了相关研究。这些新的计算模型能设计出比一般借力飞行变轨模型更好的解，但对优化算法的设计所提出的新的挑战并不多。此外，基于持续小推力变轨的交会和利用引力平衡点附近的“高速通道”进行低能转移的交会模型(龚胜平等，2007；陈方杰，2007)，以及基于这些模型的优化算法，国内外也有大量的相关研究，但这些问题本书没有展开讨论。

在多体轨道设计中，传统的求解方法是利用打靶法，将两点边值问题转化为单点初值问题进行求解。但是当目标点离目标天体较近时，打靶法就很难收敛到目标点。本书提出了一种逐次逼近打靶法，通过大量的实验表明，逐次逼近打靶法较传统打靶法有更高的收敛速度和计算精度。

在国内航天器轨道优化算法研究中，既有传统的基于牛顿迭代等确定性的局部寻优算法的研究(乔栋等，2005a)，也有基于演化算法等全局优化算法的研究(张洪波等，2006)，但这些研究多是针对特定优化模型下的算法应用研究。

演化算法作为一个新型的优化工具，在国内外已经有很多研究机构将演化算法成功地应用到了轨道优化设计中。

2003年，ESA为了增加航天任务的安全性与可靠性，降低任务的风险与成本，增加空间任务成果的回报，其下属机构ACT(Advance Concept Team)开展了空间任务分析与设计的全局优化算法研究，以解决航天器的轨道设计优化问题。研究报告对任务设计问题从问题约束、推力控制类型、航天器模型、飞行动力学模型和模拟精度来进行分类研究(Myatt，*et al*，2003)。将任务设计问题依据其复杂性由简到繁划分为：轨道优化设计问题、弱稳定边界问题和优化姿态控制问题；指出了应用全局优化的必要性和如何选择合适的全局优化算法；开发了与任务设计问题相关的全局优化新颖的可视化工具。研究报告认为，差分演化算法是解决任务设计问题最有效的方法。

2006年，ACT的Izzo等(2006)研究了多借力飞行轨道优化问题的空间修剪技术，得到一个具有多项式时间复杂度的优化算法，后被ESA称为“有里程碑意义”的研究成果。

2007年9月，Kalyanmoy(2007)、戴光明等(2009)将多目标优化思想引入深空探测任务航天器轨道优化问题中，采用多目标优化算法NSGA-Ⅱ对轨道转移时长与转移所消耗能量进行优化。同期，ACT的Tamás Vinkó等(2007)给出了特定轨道计算模型在给定约束条件下一些优化问题的Benchmark，为研究行星际轨道优化算法提供了重要的参考标准。

计算机仿真是通过建立数学模型以程序的方式来模拟真实的事物，可以对研究目标进行快速准确的分析，同时具有节约投资和直观生动的优点，计算机仿真已经广泛应用于各个领域。计算机仿真技术在航天任务轨道设计和优化的研究工作中也得到广泛的应用，为航天事业的发展和技术应用发挥着不可替代的作用。

国外许多航天机构都拥有专门用来求解轨道优化问题的软件包。比较著名的有NASA所采用的POST(Program to Optimize Simulated Trajectories)、美国航天公司研制的一套GTS(Generalized Trajectory Simulation program)、德国的BNDSCO(Numerical Solution of Optimal Control Problem)和DIRCOL(Direct Collocation)、欧空局近年来研制成功的两套大型轨道优化包ASTOS(Aerospace Trajectory Optimization Software)和ALTOS。这些软件包优化问题的求解基本上都采用直接法。

加州理工学院的喷气推进实验室提供了用于多天体交会轨道设计的工具STOUR(Satellite Tour Design Program),它是基于圆锥曲线拼接的原理。STOUR一般给出初始的设计参数,可用来进行更精确的引力场模型的优化,其缺点就是不能确保找到所有满足发射要求的轨道。为此,Longuski等人提出了一种自动设计多天体交会借力飞行轨道的方法,即给出初始发射的时间段和目标星体,通过自动寻找C3(即飞出和飞入借力天体的能量)的匹配,找出所有满足条件的发射机会。这种方法不仅有效地提高了效率,同时给出了更多的轨道设计方案,使STOUR升级为STOURA,即发射机会的自动化搜索软件,它可以完成轨道的初始方案设计。

STOURA的不足之处是不能确定轨道设计中深空机动的位置,这决定了该软件只能设计出多天体交会的纯借力飞行轨道方案。随后,Longuski等人又提出了一种自动设计带有深空机动的借力飞行轨道的方法,通过这种方法设计的深空机动点将使得在两个引力体之间飞行段的能量最小,由此他们设计出了用于任务总能量优化分析的MIDAS软件。

尽管圆锥曲线拼接法对于深空轨道初始设计与任务验证来说可以提供足够的精度,但是在实际工程中,精确模型下的轨道优化问题是不可缺少的。Sauer提出了一种基于主矢量原理的纯借力飞行轨道的优化方法,考虑了第二体引力,比圆锥曲线拼接法得到的轨道精确,但是递推比较复杂。针对带有深空机动的多天体交会借力飞行轨道,Amario和Byrnes等人提出了带有飞越高度和飞越方向约束的轨道优化方法,即通过变化借力天体之间深空机动点的位置来寻找总的深空机动的最小值,通过求解N体交会问题可以得到连接深空机动点之间的轨道段,运用多次圆锥曲线法和牛顿算法得到优化结果。该方法成功应用于伽利略探测器的轨道优化。

STK(Satellite Tool Kit)作为航天任务仿真软件的代表,具备可视化程度高、软件模块功能完善、模型精确可靠、使用方便等特点,在国内外得到了广泛和成功的应用。STK是AGI公司开发的一个综合性的软件包,除直接操作该软件进行航天任务仿真分析外,STK还提供了软件二次开发接口,为用户进行系统集成提供了支持。基于STK提供的丰富功能进行航天任务可视化仿真,可以利用STK经过验证的大量仿真模型快速、有效地构建仿真系统;丰富的三维和二维仿真显示功能则为用户提供了直观、易于理解的表现形式,从而可以更容易地发现航天任务中存在的问题(张占月等,2006)。由于界面友好等特性,STK在教学中也得到广泛应用。当然STK也有其自身的缺陷,STK主要关注仿真,而在航天任务

轨道优化方面能力不足。

我国对开发航空和航天任务轨道仿真软件进行了大量的研究。如清华大学的小卫星姿态控制系统仿真平台(徐晓云等,2003)注重于三维实时现实技术;西安导航技术研究所的飞行航迹产生软件(马云,2003)可以仿真飞机的多种飞行运动轨迹;国防科技大学的交会对接仿真系统(王华,2002)则注重于航天器的交会和对接的数值仿真;解放军信息工程大学2006年开展的导航载体轨迹仿真系统的研究与开发(郑亚弟,2006)和航天任务实时三维可视化仿真(蓝朝桢,2007)等。这些仿真系统主要专注于地球卫星任务的仿真,不具备深空探测的仿真和航天任务轨道的优化功能。

§1.2　轨道设计基本概念

航天任务设计问题可以从问题约束、控制类型、航天器模型、太阳系统模型和模拟精度来分类研究。任务设计问题依据问题复杂性由简到繁可以划分为:轨道设计问题、弱稳定边界问题和优化姿态控制(Myatt, *et al*,2003)。而本节主要讨论的就是轨道设计的基本问题。

1.2.1　基本问题

(1)点到点转移:这是一类最基本的问题,它解决的是从某一个惯性坐标系中的一个固定点到另一点的转移,这种转移通常假定转移轨道是开普勒轨道,实际上也可以不是开普勒轨道。

(2)Lambert(兰伯特)问题:如果在点到点转移问题中增加转移时间的限制条件,即从一点到另一点的时间要求在时间 t 内完成,并且转移的轨道是开普勒轨道,则这样的问题就是 Lambert 问题。这种转移通常至少需要一个单脉冲推力,一般通过数值计算的方法可获得 Lambert 问题的解,即推力的方向、大小以及转移轨道。依据天体动力学模型,Lambert 问题的解是在一个连续的二维或三维的空间中,我们将在 3.1 节中具体地讨论该问题。

(3)最小 Δv Lambert 问题:它同一般的 Lambert 问题不同,最小 Δv Lambert 问题不仅限制了转移的时间,同时要求脉冲推力最优,即 Δv 最小。

(4)轨道到轨道转移:一般是指从一个开普勒轨道转移到另外一个开普勒轨道。

(5)霍曼(Hohmann)变轨:要求双脉冲推力作用,霍曼变轨是共面圆轨道之间的最佳变轨方式,脉冲推力 Δv 也可以直接求得。霍曼变轨也可以扩展到任意椭圆形变轨,问题的解可在一维空间中搜索得到。关于霍曼变轨的详细分析将在 2.3.2 节中给出。

(6)连续推力变轨:在变轨过程中通过连续的推力作用而达到变轨的目的,因为有外力的作用,转移的轨道不是开普勒轨道。另外,最小时间连续推力变轨与连续推力变轨类似,但它是另一类变轨问题。

(7)体到体变轨:一般为行星际的轨道转移,体到体变轨是标准轨道设计问题中最复杂

的一类，最常见的就是地球-火星的转移轨道设计。由于行星的周期性运动，使得这类问题存在着多个局部极小值(全局最小值是孤立的，在 2.3.2 节我们将具体讨论)。本书将对此问题进行分析。

(8)借力飞行变轨，即引力辅助：经常被用来减少 Δv 的需求量和任务持续的时间。通过选择一些被公认的借力行星序列，使用多借力飞行来设计行星际飞行轨道。借力飞行轨道设计问题本身存在着很多局部极小值(全局最小值是伴生的)，而借力行星序列的选择引入了一些离散维变量到这类优化问题中。因此不能依靠传统的优化算法，如需要计算梯度的方法来达到全局优化的目的。另外，搜索空间的维数也因为会受到可能的借力次数和航天器的控制模型这两个因素的影响，而且事实上，绕行星借力变轨的次数也是一个优化的参数，因此该问题的搜索空间是变维的(杨嘉墀等，2005)。借力飞行轨道的设计将在 3.3 节中具体地讨论。

1.2.2 转移轨道属性描述

一般来讲，因为对问题复杂性的理论分析是极其困难的，Törn 推荐对问题根据经验的研究来决定它们的难度(Myatt，*et al*，2003)。对于问题复杂性，有 4 个属性是非常重要的：全局最优吸引盆的大小、可提供的函数评估的次数、全局最小值的伴生或孤立和局部最小值的数量。

1. 全局最优吸引盆的大小

通过在定义域中取样，并使用一个带有无穷小的步长的严格递减的局部最小值找出对应的最小值，可以得到全局最优的吸引盆的大小。这将给出吸引盆填充定义域的近似比例值。尽管这样找出的最小值在实际中是不可能的，但是所有的局部优化算法将给出一些目标函数的近似值，如 SQP 局部优化方法。

全局最小值的吸引盆的相对大小与定义域的大小成反比。因此一个最小的搜索定义域一般要使吸引盆的比例最大，同时使函数的凸性最大。

2. 可提供的函数评估的次数

使用一个随机搜索的方法或者穷举法，任何全局最优能够在无限的时间里得到，但是这肯定不是解决一般优化问题的最好方法。因此，要确定函数评估的次数 N_f，否则认为优化问题是难以处理的。很明显，这在主观上依赖于意识到的可接受的时间和可提供的计算资源，除在严格的约束条件下，N_f 值很难确定。

然而，如果与优化算法相关的消耗相比于评估目标函数的消耗是可以忽略的，那么可以有效地比较不同的优化算法所需要的函数评估的次数。

3. 伴生的或孤立的全局最小值

Törn 定义全局最小值的伴生是指在搜索空间中局部最小值出现在全局最小值附近，找出一个这样的局部最小值中的一个可以帮助找出全局最优解，这是可行的。因为任何随机

的全局最优搜索算法具有通过许多局部最小值朝着全局最小值改进的可能性。附近有其他的局部最小值的全局最小值认为是伴生的，否则是孤立的。

在两个不同的问题中全局最小值的伴生性可以用如下方式比较：首先，把搜索空间标准化为单位超立方体；然后，按照距离全局最优解的欧几里德距离，作局部最小值密度的柱状图。局部最小值的欧几里德距离以因子 $n^{1/2}$ 进一步标准化，$n^{1/2}$ 是将搜索空间规格化后得到的超立方体的对角线。

4. 局部最小值的数量

尽管全局最小值被局部最优解包围是很有好处的，然而它们减小了全局最小值的吸引盆。因此如果一个问题的局部最小值越多，它就越难被全局优化。另外，最小值很有可能与函数的谷值相遇，这表示有无穷多个局部最小值。

地球-火星转移轨道是体到体变轨问题中最常见和最基本的轨道设计问题。下面将使用以上讨论的 4 个属性来分析该问题的复杂性。

地球-火星转移轨道是一种双脉冲变轨方式（见 2.3.2 节），即离开地球时的一次脉冲加力和到达火星时的一次脉冲制动。该问题中的决策变量为发射时间 t_0 和转移时间 t，对于推力最优而言，设航天器的位置为 $\vec{R}$，则目标函数 f 等于：

$$f = |\dot{\vec{R}}_i| + |\dot{\vec{R}}_f| \tag{1-1}$$

式中：$\dot{\vec{R}}_i$ 为逃离地球时相对于地球的逃逸速度；

$\dot{\vec{R}}_f$ 为进入火星影响球时相对于火星的速度。

$\dot{\vec{R}}_i$ 和 $\dot{\vec{R}}_f$ 可以通过给定的发射时间 t_0 和飞行时间 t，求解 Lambert 问题得到。

分析地球-火星转移轨道问题时，需要考虑到行星的会合周期。因此决策变量的选取需要包含有几个会合周期，发射时间 t_0 和飞行时间 t 的范围选取在：$t_0 \in [800, 3\,800]$MJD2000（以 2000 年 1 月 1 日凌晨零点为起始时刻），$t \in [100, 400]$Days。

图 1-1 显示了在二维空间下的目标函数的分布，可以看到有多个局部极小值，并且该函数是一个类似周期性的函数，其周期取决于会合周期。如果轨道是圆形的，并且是共面，所有的局部极小值将是一样的全局最小值。因为在每一个会合周期，相对于行星的同一位置会重复出现。然而，对于非共面和非圆的轨道，存在一个唯一的全局最优值。

为了测试全局最优的吸引盆，将搜索空间每一维划分成 50 块，在每一个格网内，执行 SQP 搜索，找到对应网格的局部最小值，使用这种方法，找到 11 个不同的最小值。图 1-2 中的五角星显示其结果，并且以不同的颜色绘制出了每个最小值的吸引盆大小。

表 1-1 显示了所有对应于 SQP 算法的最小点的决策变量和目标函数的值，9 到 11 不是真正的最小点，仅仅是边界限制在决策变量上的结果，如果这些边界不是严格的，那么这些最小点不会出现。

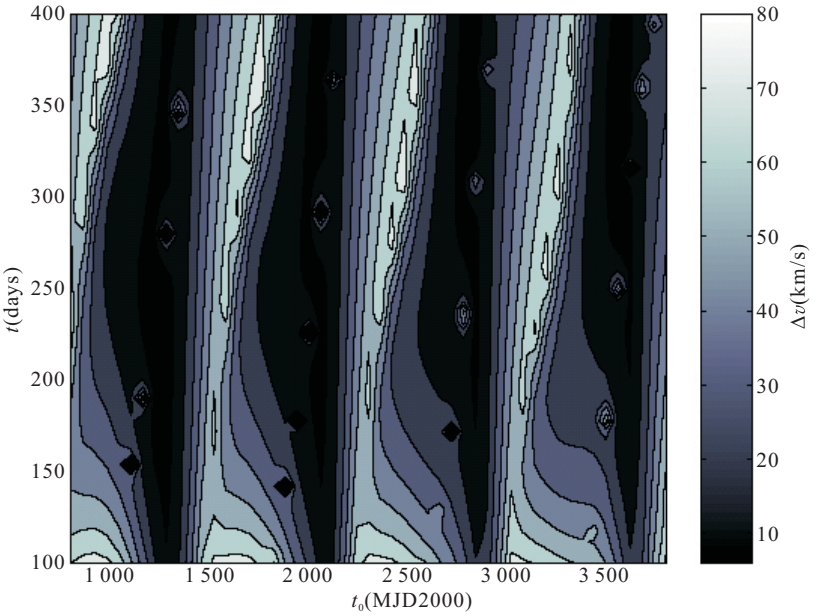


图 1－1　地球-火星最优推力问题的目标函数分布图

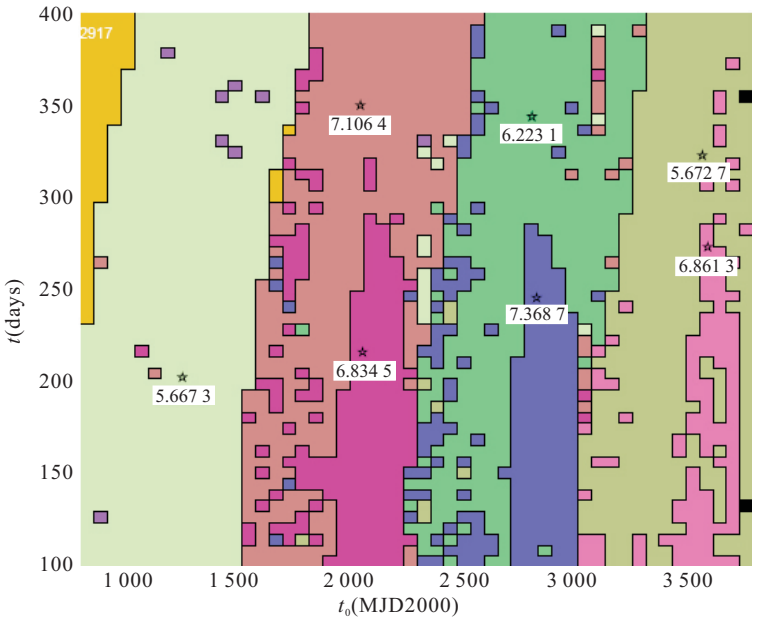


图 1－2　地球-火星推力最优转换的吸引盆

表 1－1　最小值的决策变量和目标值

最小点	t_0(MJD2000)	t(days)	$\lvert v_i \rvert$(km/s)	$\lvert v_f \rvert$(km/s)	f(km/s)
1	1 253.7	202.9	2.968	2.699	5.667
2	3 573.4	323.2	3.207	2.466	5.673
3	2 812.5	344.2	3.659	2.564	6.223
4	1 225.5	233.2	3.539	2.816	6.353
5	2 057.5	216.5	4.180	2.654	6.835
6	3 597.8	273.5	4.364	2.497	6.861
7	2 048.2	350.5	4.170	2.936	7.106
8	2 834.3	246.0	4.768	2.601	7.369
9	3 766.8	400.0	10.027	11.606	21.633
10	800.0	400.0	26.195	18.094	44.292
11	3 800.0	126.7	37.877	25.439	63.289

全局最优解吸引盆的大小

表 1－2 按比例给出了每一个最小值所占搜索空间的相对大小。很明显，全局最优(最小点 1)的吸引盆大小大约是整个搜索区域的 25%，因此，采样点不落在全局最优吸引盆的概率约为(1－25%)，经过 N 次类似搜索后，搜索点不落在全局最优吸引盆内的概率约为$(1-25\%)^N$。

表 1－2　使用 SQP 找到的近似比例的吸引盆大小

最小点	吸引盆大小	最小点	吸引盆大小
1	0.252 0	7	0.170 4
2	0.168 0	8	0.082 0
3	0.165 2	9	0.000 4
4	0.003 2	10	0.025 6
5	0.085 2	11	0.000 4
6	0.047 6		

函数评估的次数

使用表 1－2 的信息，可以估计出 SQP 搜索的次数，可以得到为了使找到全局最小值的概率达到 99％而需要评估函数的次数。如前所述，假设全局最小值的吸引盆大小与搜索域的比值为 p^*，那么，经过 K 个函数评估后，没有找到一个在全局最小值的吸引盆中的点的概率约为：

$$p_{\text{fail}} = (1 - p^*)^K \tag{1-2}$$

因此，如果需要达到一个指定的 p_{fail}，需要的采样数即函数评价的次数为：

$$K = \frac{\log(p_{\text{fail}})}{\log(1 - p^*)} \tag{1-3}$$

为了能够使找到一个或更多的全局最优的概率达到 99％，至少需要 16 次随机初始条件的 SQP 搜索。这个结果也可以根据任何给定的发射窗口得到：由于 4 个地球-火星会合周期包含在发射窗口中，并且 $p^* \approx 1/4$。这里，假设每个周期内的全局最小值是一样的，所以吸引盆的相对大小大约为 $p^* \approx 1/i$，其中 i 为包含在发射窗口中的会合周期数目。因此，函数评估的数量可以近似地表示为：

$$K \approx \frac{\log(p_{\text{fail}})}{\log(1 - \frac{1}{i})} \tag{1-4}$$

由于当 $0 < x \ll 1$ 时，$\log(1-x) \approx -x$，则式(1－3)可写成：

$$K \approx -i \cdot \log(p_{\text{fail}}) \tag{1-5}$$

所以，需要的 SQP 搜索次数与发射窗口的大小近似地呈线性关系。因此，如果对 SQP 的参数适当地初始化，那么 SQP 的计算消耗相对于会合周期的数量将保持恒定，并且，该问题的时间复杂度与发射窗口的大小成正比。

局部最优解个数

在选择的发射窗口中，找到了 10 个极小值。根据 Törn 对同样拥有 10 个极小值的“Shekel 10”函数分类，该问题只有不到 10 个极小值，因此可归类为局部极小值较少的问题。

全局最优的伴生性

图 1－3 的柱状图显示在该问题中，各个局部极小值离全局最小值的归一化距离。从图中很容易看出，没有局部最小值非常靠近全局最小值，因此，该问题的全局最小值是孤立的。

问题的复杂性分类

在这个问题中，已经确定$(1-p^*)^K$ 是很小的，最小值的个数也是比较少的，并且最小值是孤立的，因此这类问题可以被认为是容易被优化的问题。

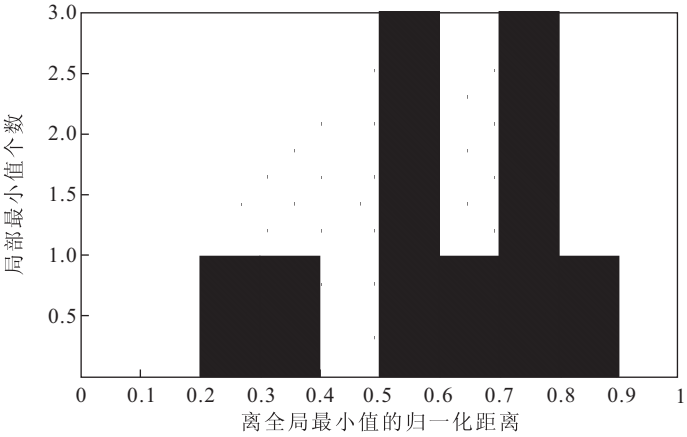


图 1-3　局部最小值离全局最小值的分布统计

§1.3　非线性规划的最优化算法

使用全局优化技术求解航天器轨道与星座优化问题，提高任务分析和设计准确性。目前在进行航天器轨道设计优化时，人们更多地采用了离散化的方法、配点法、多重打靶法和多重参数优化方法，变换最优控制问题为非线性规划问题，采用非线性规划方法求解。然而参数化后的最优控制问题往往是一类高维的非光滑问题，应用传统的非线性规划算法，比如序列二次规划方法(SQP)、共轭梯度法和单纯形算法求解该问题时，往往遇到病态梯度、初始点敏感和局部收敛等问题。特别是对于一类终端时刻可变的优化控制问题，基于梯度的非线性规划算法则是遇到了更大的麻烦。NASA 和 ESA 将近年来兴起的启发式算法、遗传算法、禁忌搜索等不完全算法应用到航天器的飞行轨道设计、姿态控制等问题中，并取得了很好的效果。

为了叙述方便，记数学规划模型的一般形式：

$$\min \quad f(x) \text{ ——目标函数}$$

$$\text{s.t. } x \in S \text{ ——约束集合，可行集}$$

其中，$S \subseteq R^n$，$f: S \to R$，$x \in S$ 称为问题的可行解。

• 若 $x^* \in S$，满足 $f(x^*) \leqslant f(x)$，$\forall x \in S$，则称 x^* 为全局的最优解(最优解)；若 x^* 为问题的最优解，则称 $f^* = f(x^*)$ 为问题的最优值(最优目标函数值)。

• 若存在 x^* 的邻域 $N(x^*)$，使得 $f(x^*) \leqslant f(x)$，$\forall x \in S \cap N(x^*)$，则称 x^* 为问题的局部最优解。

在上述定义中，当 $x \neq x^*$ 时有严格不等式成立，则分别称 x^* 为问题的严格全局最优解

和严格局部最优解。

数学规划模型建立之后，主要问题是如何求解这个数学模型。最优化问题的求解方法可以分成以下几类。

(1)间接法(或称解析法)。这种方法只适用于目标函数(或泛函)及约束有明显的解析表达式的情况。求解方法是先求出最优的必要条件，再来求解这组方程或不等式。一般是用求导数的方法或变分方法求出必要条件，通过必要条件将问题简化，因此称为间接法。主要包括：求解无约束问题的古典微分法、古典变分法；求解有约束优化问题时根据库恩-图克定理、极大值原理而形成的策略和方法。

(2)直接法。当目标函数较复杂或不能用变量显函数表示时，无法用解析法求必要条件，这时要用直接探索的方法经过若干次迭代搜索到最优点。这种方法常常是根据经验或试验得到的。对于一维搜索问题(单变量极值问题)主要是应用区间消去法(斐波那西法、黄金分割法等)或多项式插值法。对于多维搜索问题(多变量极值问题)主要应用爬山法。这类方法主要包括：坐标轮换法、步长加速法、方向加速法、单纯形法及随机搜索法。

(3)以解析法为基础的数值计算法。这类方法也是一种直接法，但它是以梯度法为基础，因此是一种解析与数值计算相结合的方法。其中有无约束梯度法(最速下降法、拟牛顿法、共轭梯度法及变尺度法等)、有约束梯度法(可行方向法、梯度投影法等)和化有约束问题为无约束问题的方法(SUMT 法、SWIFT 法和复形法等)。

(4)以演化算法为代表的随机搜索算法。由于传统的优化方法难于处理高维、非线性、目标函数或约束条件无明显解析式等问题，存在的缺点和不足主要有：

• 一般对目标函数都有较强的限制性要求，如连续、可微、单峰等。

• 大多数优化方法都是根据目标函数的局部展开性质来确定下一步搜索方向，这与求函数的整体最优解的目标有一定的抵触。

• 在实现算法之前，要进行大量的准备工作，如求函数的一阶和二阶导数，某些矩阵的逆等。在目标函数较为复杂的情况下，这项工作是很困难的，有时甚至是不可能的。

• 算法结果一般与初始值的选取有较大的关系，不同的初值可能导致不同的结果。初始值的选取较大地依赖于优化者对问题背景的认识及所掌握的知识。

• 算法缺乏简单性和通用性。针对一个问题，优化方法的使用者需要有相当的知识去判定使用哪一种方法较合适。这一困难是使优化方法得到更广泛应用的主要障碍之一。

• 对某些约束问题较难处理。

近年来，演化计算等基于自然法则的随机搜索算法，在优化领域取得的成功引起了人们的普遍关注。越来越多的人加入到演化优化的研究之中，并对演化算法作了诸多改进，使其更适合优化问题。同时，对演化计算所适合的优化问题的讨论也是目前研究的热点。近年来，人们发展了一类以演化算法为核心的随机搜索算法。

第二章　行星际轨道动力学基础

在航天器动力学中，把研究两个质点模型天体在它们之间的万有引力作用下的运动问题称为二体问题，即忽略其他天体的作用，只研究某一个影响力最大的天体对航天器的作用；把航天器在多个天体的万有引力共同作用下的运动问题称为 N 体问题。二体问题可以得到形式简单的解析解，而多体问题只能依靠数值积分方法计算得到。本章介绍航天器的轨道动力学基础知识。

§2.1　常用时空系统

时间离不开物质的运动，要研究航天器在太空中的运动规律，必须确定参考系。在参考系中通过时间的计量来研究航天器的运动规律。

2.1.1　常用时间系统

研究物体运动，需要一种均匀的时间尺度。历史上，这种均匀的时间尺度是以地球自转为基础的。但由于地球自转的不均匀性，这种时间系统已经不适应；但由于地球自转与人们的生产活动密切相关，因此时间系统又必须与地球自转相协调，这就导致了时间系统的复杂化。

现行的时间系统基本上分为（郗晓宁等，2003）：以地球自转运动为依据建立的时间系统，称为世界时；一类是以地球公转运动为依据建立的时间系统，称为历书时（历书时系统由于存在诸多缺陷，从 1984 年开始，已完全被原子时所代替）（马文臻，2006）。

原子时秒长均匀，稳定性强，但它却与地球自转无关，在研究航天器运动规律时，却涉及到计算地球的瞬时位置等数据，这又需要世界时。因此为了兼顾世界时和原子时的秒长，建立一种折衷的时间系统，称为协调世界时 UTC（Universal Time Coordinated）。根据国际规定，协调世界时的秒长与原子时秒长一致，在时刻上则要求尽量与世界时接近。从 1972 年起，规定两者的差值保持在±0.9 秒内。为此可能在年底或年中对世界时的时刻作一整秒的调整，也就是通常所说的闰秒现象。最近的一次闰秒发生在 2006 年 1 月 1 日 0 点。有关闰秒的详细资料参考：http://en.wikipedia.org/wiki/Leap_second。

另外，天文上常用年的小数表示某一特殊时刻，称为历元。天文上通常有两种历元取法

和年的长度计算方法。一种叫做贝塞尔(Bessel)年,其长度是平回归年,即 365.242 198 8 平太阳日,常用的贝塞尔历元,是指太阳平黄经 280°的时刻,例如 1 950.0 并不是人们习惯的世界时 1950 年 1 月 1 日 0 时,而是 1949 年 12 月 31 日 $22^h09^m42^s$;一种称为儒略(Julian)年,其长度是 365.25 平太阳日。儒略历元 2 000.0 即 2000 年 1 月 1 日 $12^h00^m00^s$。儒略日计日法可以倒退到公元前 4713 年 1 月 1 日的格林尼治(Greenwich)平午(即世界时 12^h),每天顺数而下。显然,引用儒略年比贝塞尔年更方便在人们习惯使用的世界时之间进行转换,因此从 1984 年起,贝塞尔年已被儒略年所代替。

随着时间的推移,儒略日数值已经很大,因此引入约简儒略日,记为 MJD(Modifier Julian Date),它从世界时 1858 年 11 月 17 日 0 时开始。定义为:

$$\mathrm{MJD}=\mathrm{JD}-2\ 400\ 000.5 \tag{2-1}$$

MJD 自 1858 年 11 月 17 日 0^hUT 开始。另一个修正儒略历元为 MJD2000,定义为:

$$\mathrm{MJD2000}=\mathrm{JD}-2\ 451\ 544.5 \tag{2-2}$$

即世界时 2000 年 1 月 1 日 0 时为 MJD2000 的起始时刻。

由于儒略日的计时方法是基于世界时的,且它与人们习惯的年月日表现形式不一样。因此通常需要在儒略日与世界时之间进行转换。

设 UT/UTC 日期格式为 Year-Month-Day,给出转换公式如下:

(1) UT/UTC 日期转换成儒略日。

$$\begin{aligned}\mathrm{JD}=\mathrm{Day}-32\ 075&+\left\{\frac{1\ 461}{4}\times\left[\mathrm{Year}+4\ 800+\left(\frac{\mathrm{Month}-14}{12}\right)\right]\right\}\\&+\left\{\frac{367}{12}\times\left[\mathrm{Month}-2-\left(\frac{\mathrm{Month}-14}{12}\right)\times 12\right]\right\}\\&+\left\{\frac{3}{4}\times\left[\mathrm{Year}+4\ 900+\frac{1}{100}\times\left(\frac{\mathrm{Month}-14}{12}\right)\right]\right\}-0.5\end{aligned} \tag{2-3}$$

(2)儒略日转换成 UT/UTC 日期。

$$\begin{cases}J=\mathrm{JD}+0.5\\N=4\times(J+68\ 569)\div 146\ 097\\L_1=J+68\ 569-[(N\times 146\ 097+3)\div 4]\\Y_1=4\ 000\times(L_1+1)\div 1\ 461\ 001\\L_2=L_1-[(1\ 461\times Y_1)\div 4]+31\\M_1=80\times L_2\div 2\ 447\\L_3=M_1\div 11\\\mathrm{Day}=L_2-[(2\ 447\times M_1)\div 80]\\\mathrm{Month}=M_1+2-12\times L_3\\\mathrm{Year}=100\times(N-49)+Y_1+L_3\end{cases} \tag{2-4}$$

儒略日的小数部分由世界时的时分秒(Hour－Minute－Second)换算,公式如下:

$$\text{Fraction}=\frac{\text{Hour}}{24}+\frac{\text{Minute}}{1\ 440}+\frac{\text{Second}}{86\ 400} \tag{2-5}$$

另外,由于世界时是不均匀的时间尺度,因此在计算精度要求很高时,就需要使用基于原子时的时间系统。这就导致虽然我们可以按式(2－3)～式(2－5)很容易的在 UT/UTC 和儒略日(JD)之间进行转换,但在计算两个儒略日之间精确时间间隔时,则需要考虑闰秒的问题。简单的方法是将两个儒略日转换为国际原子时(International Atomic Time,TAI)秒,再进行秒差的计算。表 2－1 给出了 1972 年至今的闰秒记录。

表 2－1　闰秒记录

TAI－UTC(秒)	发生闰秒的时刻(JD)	发生闰秒的时刻(UTC)
10	2 441 317.5	1972 年 1 月 1 日 0 时
11	2 441 499.5	1972 年 7 月 1 日 0 时
12	2 441 683.5	1973 年 1 月 1 日 0 时
13	2 442 048.5	1974 年 1 月 1 日 0 时
14	2 442 413.5	1975 年 1 月 1 日 0 时
15	2 442 778.5	1976 年 1 月 1 日 0 时
16	2 443 144.5	1977 年 1 月 1 日 0 时
17	2 443 509.5	1978 年 1 月 1 日 0 时
18	2 443 874.5	1979 年 1 月 1 日 0 时
19	2 444 239.5	1980 年 1 月 1 日 0 时
20	2 444 786.5	1981 年 7 月 1 日 0 时
21	2 445 151.5	1982 年 7 月 1 日 0 时
22	2 445 516.5	1983 年 7 月 1 日 0 时
23	2 446 247.5	1985 年 7 月 1 日 0 时
24	2 447 161.5	1990 年 1 月 1 日 0 时
25	2 447 892.5	1991 年 1 月 1 日 0 时
26	2 448 257.5	1992 年 7 月 1 日 0 时
27	2 448 804.5	1993 年 7 月 1 日 0 时
28	2 449 169.5	1994 年 7 月 1 日 0 时
29	2 449 534.5	1996 年 1 月 1 日 0 时
30	2 450 083.5	1997 年 7 月 1 日 0 时
31	2 450 630.5	1999 年 1 月 1 日 0 时
32	2 451 179.5	2006 年 1 月 1 日 0 时
33	2 453 736.5	2009 年 1 月 1 日 0 时

UTC 转换为 TAI 的过程描述如下：

```
utcJD=UtcDateToJD(year,month,day)
for i=33 down to 0
    if utcJD>=LeapSecondTable[i][1] then break;
end for
tai=utcJD*86400+Fraction(hour,minute,second)+LeapSecondTable[i][0]
```

其中 UtcDateToJD(year, month, day) 即式(2-3)，Fraction(hour, minute, second)为式(2-5)，LeapSecondTable[i][j]表示表 2-1 的第 i 行第 j 列($0\leqslant i\leqslant 32$, $0\leqslant j\leqslant 1$)。

2.1.2 常用坐标系统

行星际转移轨道主要研究日心过渡轨道段，通常选用日心坐标系统。常用日心坐标系有日心黄道坐标系和日心球面黄道坐标系。

1. 日心黄道坐标系 $O_sx_sy_sz_s$

图 2-1 为日心黄道坐标系 $O_sx_sy_sz_s$。坐标轴 O_sx_s 在黄道面内，指向春分点；O_sz_s 轴垂直于黄道面，与地球公转角速度矢量一致；O_sy_s 轴与 O_sz_s 轴垂直，且 $O_sx_sy_sz_s$ 构成右手直角坐标系。

2. 日心球面黄道坐标系

如图 2-1 所示，日心球面黄道坐标系的三个坐标是 M、β 和 θ。M 为日心 O_s 到空间某点 N 的距离；β 在黄道面内，为春分点向东到 N 点的矢径在黄道面内投影的角距，通常称为黄经；θ 为 N 点的矢径与黄道面的夹角，通常称为黄纬。

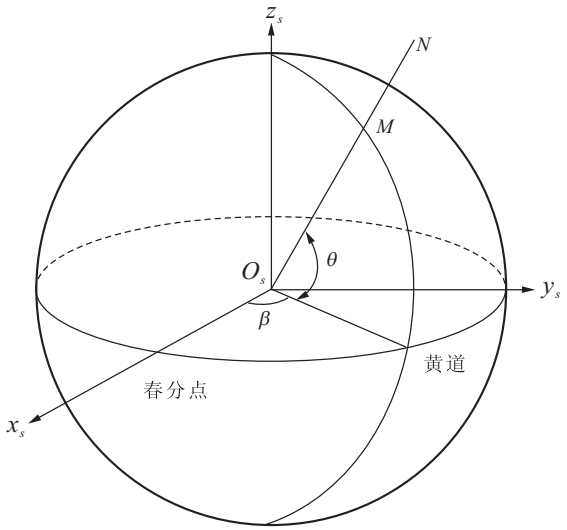


图 2-1 日心黄道坐标系 $O_sx_sy_sz_s$

§2.2　航天器动力学原理

为了研究工作和实际应用的方便，通常把作用于航天器上的各种力按其影响的大小分为两类：一类是假设引力源是一个质点，其引力称为中心力，决定着航天器运动的基本规律和特征，由此决定的航天器轨道可视为理想轨道，是分析卫星实际轨道的基础。另一类是摄动力或非中心力，包括其他星体的引力作用、太阳光辐射压力等。摄动力使卫星的运动产生一些小的附加变化而偏离理想轨道，同时偏离量的大小也随时间而改变。

在摄动力作用下的航天器运动称为受摄运动，相应的卫星轨道称为受摄轨道。

本书的重点是行星际转移轨道的初轨设计与优化，因此本项工作主要是在无摄运动的基础上进行的。

2.2.1　二体问题的解析解和轨道根数

日心黄道惯性坐标系中，在质点模型基础上，根据万有引力定律，航天器的引力加速度为：

$$\frac{\partial^2 \vec{r}}{\partial t^2} = -\frac{G(M+m)}{r^3}\vec{r} \tag{2-6}$$

式中：G 为万有引力常数；

M 为太阳质量；

m 为航天器质量；

$\vec{r}$ 为航天器在日心黄道坐标系中的矢径，$r=|\vec{r}|$。

根据式(2-6)来研究航天器的运动规律，在天体力学中称为二体问题。引力加速度决定了航天器飞行的基本规律。卫星在上述引力场中的无摄运动，也称开普勒运动，其运动规律与开普勒三大定律(马文臻，2006)描述行星运动的规律相似。

在一般情况下，对某一个天体的运动起主导作用的力，是另一个质量更大、相距较近的天体对它的吸引力。比如在太阳系内，按照引力性质就可以划分为行星空间和星际空间。在星际空间主要是以太阳的引力为主来计算的，而人造卫星轨道则是以地球的行星空间为主来计算的。在初步的分析中，往往把天体运动简化并抽象为两个质点 m、M(位于天体质心)在相互引力作用下的运动。

在日心黄道惯性坐标系中，设 M 到 m 的矢径为 $\vec{r}$，则 m 运动的加速度为 $\vec{r}$ 对时间的二阶微分即式(2-6)左边项，将其简写为 $\ddot{\vec{r}}$，另外由于太阳质量远大于航天器质量，因此式(2-6)中的$(M+m)$可近似为 M，又可记作如下形式：

$$\ddot{\vec{r}} = -\frac{\mu}{r^3}\vec{r} \tag{2-7}$$

式中：μ 为太阳引力常数，$\mu=G\cdot M$（G 为万有引力常数，M 为太阳质量）。

在航天器与太阳构成的二体系统中，航天器与太阳均被视为质点，惯性坐标系 $O_s x_s y_s z_s$ 的原点即太阳质心，航天器的运动方程可描述如下：

$$\begin{cases}\ddot{x}+\dfrac{\mu}{r^3}x=0\\ \ddot{y}+\dfrac{\mu}{r^3}y=0\\ \ddot{z}+\dfrac{\mu}{r^3}z=0\end{cases}\tag{2-8}$$

式中：x、y、z 为航天器在 $O_s x_s y_s z_s$ 中的位置矢量 3 个轴上的分量；

r 为矢量的模长，计算公式如下：

$$\text{Fraction}=\frac{\text{Hour}}{24}+\frac{\text{Minute}}{1\ 440}+\frac{\text{Second}}{86\ 400}\tag{2-9}$$

式(2-8)是一个 6 阶的非线性微分方程，如给定了 6 个初始条件：t_0 时刻航天器的位置 $x(t_0)$、$y(t_0)$、$z(t_0)$ 和速度 $\dot{x}(t_0)$、$\dot{y}(t_0)$、$\dot{z}(t_0)$，则方程组完全可解。这些初始条件确定了 6 个积分常数，每个积分常数都描述航天器的轨道、位置和航天器再次通过该时刻点的轨道周期（房建成等，2006）。式(2-8)中，第二个方程乘以 z 减去第三个方程乘以 y，可以得到：

$$\ddot{y}z-\ddot{z}y=0\tag{2-10}$$

即：

$$\frac{\mathrm{d}}{\mathrm{d}t}(\dot{y}z-\dot{z}y)=0\tag{2-11}$$

对式(2-11)积分得：

$$\dot{y}z-\dot{z}y=A\tag{2-12}$$

同理得出：

$$\begin{aligned}\dot{x}y-\dot{y}x&=B\\ \dot{z}x-\dot{x}z&=C\end{aligned}\tag{2-13}$$

进而得出：

$$Ax+By+Cz=0\tag{2-14}$$

式中：A、B、C 为积分常数，其中两个是独立的。

式(2-14)表明，航天器在一个平面内运动，这个平面通过日心，通常称这个平面为轨道平面。

航天器在轨道平面内运动，仍然满足万有引力和牛顿第二定律，其平面内方程可写为：

$$\begin{cases}\dfrac{\mathrm{d}^2\xi}{\mathrm{d}t^2}+\dfrac{\mu\xi}{r^3}=0\\ \dfrac{\mathrm{d}\eta}{\mathrm{d}t^2}+\dfrac{\mu\eta}{r^3}=0\end{cases}\tag{2-15}$$

ξ 和 η 是在 $O\xi\eta$ 坐标系(原点为日心 O,$O\xi$ 轴和 $O\eta$ 轴在卫星轨道平面内,且相互垂直)中的卫星坐标。作极坐标变换 $\xi=r\cos\theta$,$\eta=r\sin\theta$,带入式(2-15),可得:

$$\begin{cases}\ddot{r}-r\dot{\theta}^2=-\dfrac{\mu}{r^2}\\ r\ddot{\theta}-2\dot{r}\dot{\theta}=0\end{cases}\tag{2-16}$$

上面第二式可直接积分,得到:

$$r^2\dot{\theta}=h\tag{2-17}$$

h 作为积分常数,做 $1/r=u$ 的变换,并以 θ 为自变量,可得:

$$\frac{\mathrm{d}^2u}{\mathrm{d}\theta^2}+u=\frac{\mu}{h^2}\tag{2-18}$$

方程(2-18)的一般解为:

$$r=\frac{1}{u}=\frac{h^2/\mu}{1+e\cos(\theta-\omega)}\tag{2-19}$$

令 $p=h^2/\mu$,有:

$$r=\frac{p}{1+e\cos(\theta-\omega)}\tag{2-20}$$

式(2-19)为圆锥曲线方程,在这个式子中,有两个积分常量,即 e 和 ω。e 称为圆锥曲线偏心率,航天器绕大质量引力体运行时,不同的偏心率对应不同类型轨道:

当 $e=0$ 时,航天器的运动轨迹为以引力体质心为中心的圆;

当 $0<e<1$ 时,运动轨迹为椭圆,引力体处于椭圆的一个焦点上,e 为椭圆的偏心率;

当 $e=1$ 时,航天器运动轨迹为抛物线;

当 $e>1$ 时,航天器运动轨迹为双曲线。

ω 也是一个积分常量,当 $\theta=\omega$ 时,航天器距离引力体质心的距离 r 最小,该点称为近心点;当 $\theta-\omega=180°$时,航天器距离引力体质心最大,该点称为远心点。

2.2.2　轨道根数及其几何意义

描述卫星轨道位置和状态的参数称为轨道参数,也称为轨道根数,或轨道要素。确定航天器的轨道需要 6 个参数,分别是轨道倾角 i、轨道半长轴 a、偏心率 e、升交点赤经 Ω、近心点幅角 ω 和真近点角 θ。这些参数确定轨道平面在空间的指向、航天器在轨道平面中的运动方向(通常将扫过角度大于 π 的轨道称为“长程”反之为“短程”轨道)、轨道的形状和航天器所在轨道上的位置。6 个参数意义描述如下(图 2-2)。

(1)轨道倾角 i:赤道平面与轨道平面的夹角。

(2)半长轴 a:确定轨道大小的参数。对于椭圆轨道而言,就是椭圆轨道的半长轴。

(3)偏心率 e:确定轨道形状的参数。对于椭圆轨道而言,就是椭圆轨道的偏心率。

(4)升交点赤经 Ω:由春分点沿着赤道至升交点(航天器由南半球至北半球穿过赤道平

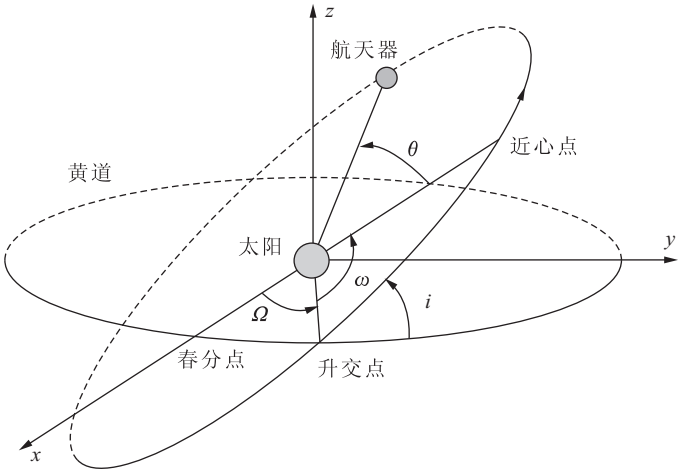


图 2-2 轨道根数示意图

面的点)的角度。

(5)近心点幅角 ω:自轨道升交点,在轨道平面内沿航天器运动方向度量至近心点的角度,即近心点矢径延长线与节线之间的夹角。

(6)真近点角 θ:自近心点沿航天器运动方向度量至航天器某时刻所在位置的角度,是随着时间不停变化的。

卫星轨道的形状和大小由轨道的偏心率 e 和半长轴 a 确定,轨道平面相对中心引力体的关系可由轨道倾角 i 和升交点赤经 Ω 确定,轨道在轨道平面中的取向由近心点幅角 ω 确定,卫星在轨道上的位置由轨道的真近点角 θ 确定,有时也使用过近心点时间 t_0 或平近点角 M[将卫星的运动轨迹虚拟为圆轨道,平近点角就是虚拟圆上的位置点、近心点、坐标原点(引力体质心)三者在原点形成的夹角]确定。

2.2.3 开普勒方程

由 2.2.1 节可知,已知某一时刻的轨道初始状态,即可求解轨道根数,并确定轨道在三维空间中的形状和方向等信息,但如何在给定轨道根数后得出任意时刻航天器所在位置呢?这一问题需要通过求解开普勒方程来解决。

为简单起见,考虑二维轨道平面,它在 Fxy 平面上。以椭圆轨道为例。

在图 2-3 中:O 为椭圆中心;F 为椭圆焦点,坐标原点;E 为偏近点角;f 为真近点角。

从图 2-3 中不难看出,若已知位置矢量 $\vec{r} = \overrightarrow{FS}$,真近点角 f,则有:

$$FR = OR - O = a\cos E - ae \tag{2-21}$$

而:

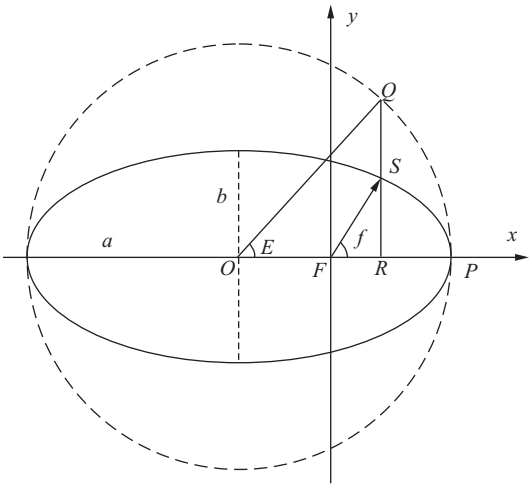


图 2-3　二体椭圆轨道示意图

$$FR = r\cos f \tag{2-22}$$

由式(2-21)和式(2-22)可得：

$$x = r\cos f = a(\cos E - e) \tag{2-23}$$

根据椭圆的性质，又可得：

$$y = r\sin f = a\sqrt{1-e^2}\sin E \tag{2-24}$$

要求椭圆轨道上某点在平面坐标中的位置，只需要知道偏近点角 E 即可。开普勒方程，就是在偏近点角 E 和时间 t 之间建立的一个函数关系：

$$E - \sin E = t\sqrt{\mu/a^3} \tag{2-25}$$

方程中的 μ 为中心引力体的引力常数，a 为轨道根数半长轴。该方程是一个超越方程需要迭代求解，具体迭代方法这里不作展开，这里引出几个结论性公式。

平面内椭圆轨道位置速度公式：

$$\begin{aligned}\vec{r} &= \{a(\cos E - e), a\sqrt{1-e^2}\sin E, 0\}\\ \dot{\vec{r}} &= \{\frac{-a^2 n}{r}\sin E, \frac{a^2 n}{r}\sqrt{1-e^2}\cos E, 0\}\end{aligned} \tag{2-26}$$

平面内双曲线轨道的位置和速度公式：

$$\begin{aligned}\vec{r} &= \{a(e - \cosh E), a\sqrt{e^2-1}\sinh E, 0\}\\ \dot{\vec{r}} &= \{-\frac{a^2 n}{r}\sinh E, \frac{a^2 n}{r}\sqrt{e^2-1}\cosh E, 0\}\end{aligned} \tag{2-27}$$

式中：n 为平均角速度，可由二体问题中心引力体引力常数 μ 和轨道半长轴求出。即：

$$n=\sqrt{\frac{\mu}{a^3}} \tag{2-28}$$

以上都是 xy 平面内的轨道位置和速度计算方法，故所有的 z 坐标值都为 0，若是三维空间中的轨道，只需将二维空间坐标点绕各轴旋转特定角度即可，旋转次序为：绕 z 轴旋转近心点幅角 ω，绕 x 轴旋转轨道倾角 i，最后绕 z 轴旋转升交点赤径 Ω。旋转公式见附录 A。

2.2.4 多体问题的运动方程

在某一惯性坐标系中，有多个（设 N 个，$N\geqslant 3$）质点（天体，虽然日心黄道坐标系中天体不能严格地看成质点，但是由于天体之间的距离很远，所以可以视为质点）的运动问题称为多体问题。航天器的质量相对于其他引力体是非常小的，因此可以略去它对引力体的影响。解决多体问题在数学上是很困难的，还没有解析解，只能用数值积分法求解。即使是三体问题，也只有在假设一些理想条件的前提下才有解析解。

设在惯性坐标系 $Oxyz$ 中，有 N 个质量分别为 $m_i(i=1,2,\cdots,N)$ 的质点构成的 N 体系统。N 个质点的位置矢量分别为 $\vec{r}_i(i=1,2,\cdots,N)$，如图 2-4 所示。

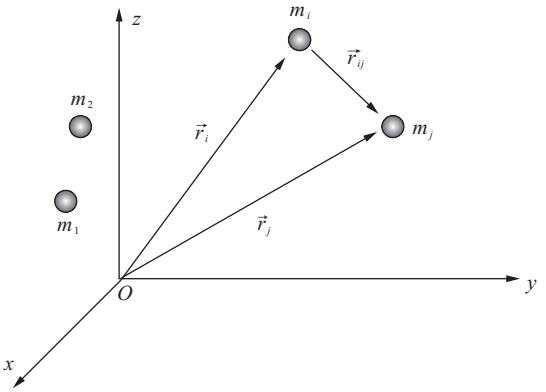


图 2-4 多体问题示意图

令 m_i 到 m_j 的位置矢量为 $\vec{r}_{ij}$，则：

$$\vec{r}_{ij}=\vec{r}_j-\vec{r}_i \tag{2-29}$$

根据万有引力定律，m_j 作用在 m_i 上的力为：

$$\vec{F}_i=\frac{Gm_im_j}{r_{ij}^3}\vec{r}_{ij} \tag{2-30}$$

作用在 m_i 上的所有力的矢量和 $\vec{F}$ 为：

$$\vec{F} = \sum_{j=1, j\neq i}^{N} \frac{G m_i m_j}{r_{ij}^{\ 3}} \vec{r}_{ij} \tag{2-31}$$

式中：r_{ij} 为 $\vec{r}_{ij}$ 的模，$r_{ij} = |\vec{r}_{ij}| = |\vec{r}_j - \vec{r}_i|$；

G 为万有引力常数。

根据牛顿第二定律可得 m_i 的运动方程为：

$$\ddot{\vec{r}}_i = \frac{\vec{E}}{m_i} = \sum_{j=1, j\neq i}^{N} \frac{G m_j}{r_{ij}^{\ 3}} \vec{r}_{ij} \tag{2-32}$$

假设 m_1 为太阳，m_2 为航天器，m_3，m_4，…，m_n 为太阳系中各大行星，那么当 $i=1$ 时，运动方程(2-32)可以写成：

$$\ddot{\vec{r}}_i = \sum_{j=2}^{N} \frac{G m_j}{r_{1j}^3} \vec{r}_{1j} \tag{2-33}$$

航天器(当 $i=2$ 时)，运动方程为：

$$\ddot{\vec{r}}_i = \sum_{j=1, j\neq 2}^{N} \frac{G m_j}{r_{2j}^3} \vec{r}_{2j} \tag{2-34}$$

航天器相对于太阳的运动为：

$$\vec{r}_{12} = \vec{r}_2 - \vec{r}_1 \tag{2-35}$$

将式(2-35)求导得：

$$\ddot{\vec{r}}_{i12} = \ddot{\vec{r}}_{i2} - \ddot{\vec{r}}_{i1} \tag{2-36}$$

再将式(2-33)和式(2-34)带入式(2-36)，化简后得：

$$\ddot{\vec{r}}_i = -G \frac{m_1 + m_2}{r_{12}^3} \vec{r}_{12} + \sum_{j=3}^{N} G m_j \left(\frac{\vec{r}_{2j}}{r_{2j}^3} - \frac{\vec{r}_{1j}}{r_{1j}^3} \right) \tag{2-37}$$

式(2-37)即为多体问题中航天器在日心坐标系中的运动方程。多体问题至今还没有解析的解法，只能依靠数值积分的方法进行计算(刘暾等，2003)，常用的数值积分方法为 Runge-Kutta 法。在日心黄道坐标系中给出任意时刻行星的位置的前提下，给出航天器的初始状态(指定时刻的初始位置和初始速度)，即可通过对运动方程进行数值积分，计算出航天器的运行轨迹。

目前，多体积分轨道主要用于数值仿真，或结合打靶法或微分校正等方法，在精确轨道确定方面广泛采用。另外由于数值积分计算量大，初轨设计的精度要求不高等因素，初轨设计中普遍都没有采用数值方法进行优化。

本节介绍了二体模型下航天器的运动方程，如何根据给定的初始状态获得轨道根数、轨道根数的几何意义，以及多体模型的运动方程。二体模型下，已知轨道根数可以确定任意时刻航天器的位置和速度，这就需要求解开普勒方程(Bate，*et al*，1990)，多体模型下则需要使用数值积分的方法进行求解。

§2.3 轨道机动和变轨

航天器从初始轨道(或停泊轨道)向目标轨道的过渡,就是一种轨道转移,它由轨道机动来完成,即变轨。初始轨道和目标轨道可以分别是圆轨道、椭圆轨道,甚至是双曲线轨道,它们之间可以是共面的,也可以是不共面的、相交的或不相交的。转移轨道可以是椭圆轨道,为了节省时间也可以是双曲线轨道。在轨道过渡中,如何选择适当的转移轨道,往往是寻求能量最省的过渡这样一个理论问题。

轨道机动按其目的不同,可分为三类(刘暾等,2003):①轨道过渡机动,又称变轨,是改变轨道参数以便从初始轨道过渡到中间轨道或最终轨道的过程;②轨道校正机动,又称轨道修正,其目的是补偿轨道参数中的误差或由各类干扰因素引起的偏差,它与轨道过渡的不同点在于它不过渡到其他轨道上去;③接近机动,即两个航天器在接近和对接过程中的轨道机动。本书所研究的是第一种机动类型,即轨道过渡机动。

轨道机动按机动力的不同,又可分为:脉冲机动和小推力持续机动。航天器发动机在短时间内(相对整个飞行过程时长来说,时间短到可以忽略)进行推进,可视为瞬间改变航天器运行速度,对应的变轨方式称为脉冲变轨;使用持续小推力机动的航天器推进动力来自飞行过程中受到的小的持续推力如离子推进器和太阳光帆等,利用这种机动方式进行变轨称为小推力变轨。

行星际转移轨道通常可利用高速运行的行星,来进行借力飞行变轨(Sim,*et al*,1996);还可以利用天体之间的引力平衡点的特殊性质进行低能耗变轨(刘林等,2005;龚胜平等,2007)。脉冲变轨和小推力变轨都可以结合借力飞行变轨来完成行星探测轨道设计任务。本书主要研究的是脉冲机动结合借力飞行变轨方式的轨道优化设计,因此有必要对这两种变轨模型进行进一步介绍。

2.3.1 单脉冲变轨

单脉冲变轨是轨道射入问题。已知卫星在某一时刻 t_0 的位置 $\vec{r}_0$ 和速度 $\vec{v}_0$,在该时刻施加脉冲式速度增量 $\Delta\vec{v}$ 之后,卫星进入另外一条轨道,其轨道要素直接由轨道射入参数 $\vec{r}_0$、$\vec{v}_0$、$\Delta\vec{v}$确定。如图 2-5 和图 2-6 分别是共面和不共面变轨,下标“0”为脉冲变轨时刻卫星的运动参数,下标“1”为脉冲作用后新轨道的运动参数和轨道要素(郗晓宁等,2003)。

2.3.2 双脉冲变轨

单脉冲变轨的主要特点是新轨道必定与原轨道相交,双脉冲变轨能使新轨道完全脱离原轨道。在两个共面圆轨道之间的最佳变轨方式为霍曼变轨;在两个圆轨道之间的最佳过渡轨道为霍曼椭圆,此椭圆分别与两个圆轨道相切,切点即为过渡轨道的近地点和远地点,

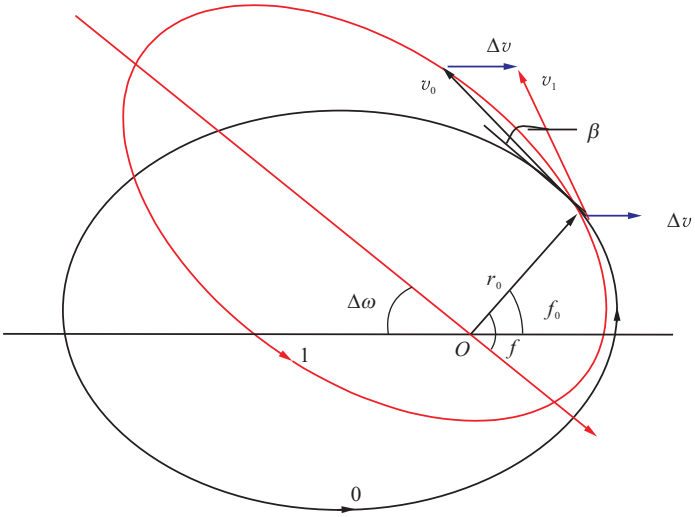


图 2－5　单脉冲变轨示意图——共面

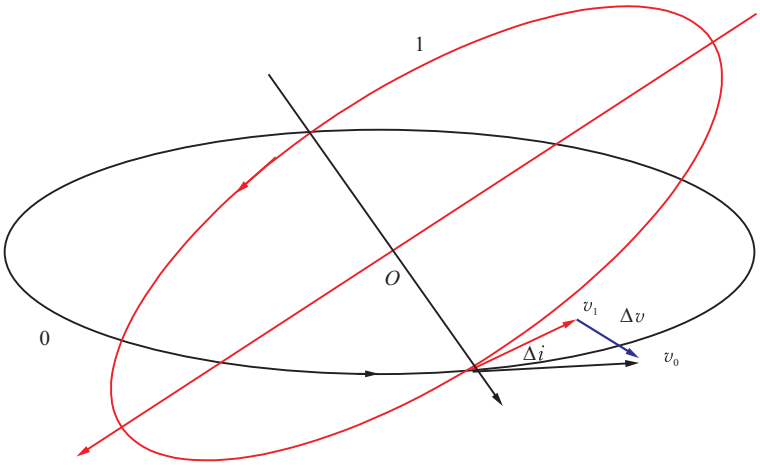


图 2－6　单脉冲变轨示意图——不共面

如图 2－7 所示。

考虑一种简单的轨道过渡：初始轨道和目标轨道为两个同心圆（中心引力体的质心）轨道，见图 2－7 中轨道 A 和轨道 B 各为低圆轨道和高圆轨道，无论是从低圆轨道过渡到高圆轨道，还是从高圆轨道过渡到低圆轨道，在限定只用两次脉冲推力的情况下，采用与两个圆轨道相切的椭圆轨道 H 过渡，耗费能量最小。这是霍曼在 1925 年首次提出的，人们称其为霍曼（Hohmann）变轨，相应的转移轨道称为霍曼转移轨道。

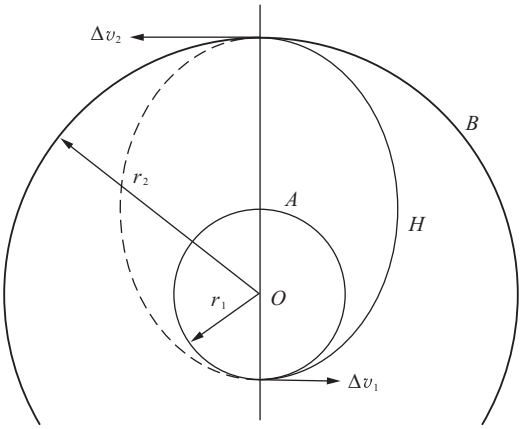


图 2-7 霍曼转移轨道示意图

上述过渡事实上就是两次改变轨道半长轴的转移过程。两个圆轨道的半长轴各为 r_1 和 r_2，如果从低圆轨道（初始轨道）向高圆轨道（目标轨道）过渡，霍曼转移轨道的半长轴 a 应变为：

$$a = \frac{1}{2}(r_1 + r_2) \tag{2-38}$$

由此不难得出两次变轨前后的半长轴变量。根据半长轴变量，可求得两次脉冲大小如下（刘林等，2005）：

$$\Delta v_1 = \sqrt{\frac{\mu}{r_1}}\left[\left(\frac{2r_2}{r_1 + r_2}\right)^{1/2} - 1\right] \tag{2-39}$$

$$\Delta v_2 = \sqrt{\frac{\mu}{r_2}}\left[1 - \left(\frac{2r_1}{r_1 + r_2}\right)^{1/2}\right] \tag{2-40}$$

霍曼轨道虽然是理想前提下的转移轨道，但其最优性（证明见耿长福，2006）却对实际工程和应用有很大的参考价值。实际上，近似理想条件下的双脉冲转移轨道（如地球-火星转移轨道）的最优解也与霍曼轨道相近。

另外一种最佳脉冲共面变轨是双椭圆三脉冲方式。当外圆半径大于 11.98 倍内圆半径时，三脉冲比霍曼变轨更有利。

霍曼变轨原则可推广到非共面圆轨道之间的变轨。霍曼椭圆过渡轨道的主轴与两轨道平面的相交节线一致，椭圆的近、远地点分别与两轨道相接，椭圆过渡轨道平面介于两平面之间，两次脉冲不仅改变轨道的长半轴，同时起到改变轨道平面的作用。

共面椭圆轨道的变轨是更为一般性的轨道控制，其主要内容是椭圆拱线转动控制，另外两个独立变量偏心率和半长轴可保持定值或按照相应要求变化。双脉冲拱线转动控制是共

面椭圆变轨的一种最佳控制模式，其简化模型是两脉冲 180°对称周向控制（郗晓宁等，2003）。

2.3.3　借力飞行变轨

借力飞行变轨（Gravity Assist，缩写 GA，又称 Swing - by，Fly - by，借力飞行等）。常被用于减少深空探测任务燃料消耗（Δv），缩短飞行时长（Δt）。为了更直观地理解借力飞行变轨的物理原理，罗治情（2007）称之为“弹弓效应”的现象，并结合图解，形象地描述了为什么借力飞行变轨可以改变航天器的速度大小和方向。下面对借力飞行变轨的求解模型作简单介绍。

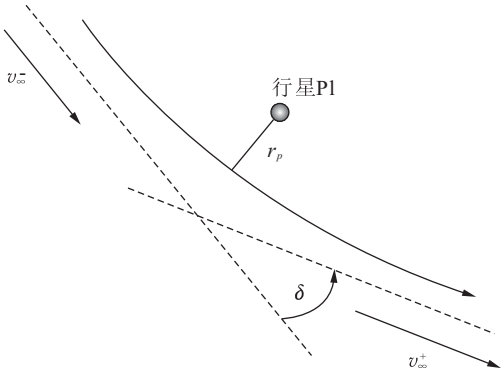


图 2 - 8　借力飞行示意图

图 2 - 8 表明了航天器掠过行星进行借力飞行变轨的原理。航天器相对于绕飞行星的飞行轨道是一条双曲线飞越轨道，相对行星的渐近线速度是 $\vec{v}_\infty$。相对行星而言，借力飞行的作用仅仅是旋转 $\vec{v}_\infty$ 的方向，而不改变其大小（即 $|\vec{v}_\infty^-| = |\vec{v}_\infty^+|$），使速度从飞入影响球（影响球是行星引力作用的有效范围，详细内容将在第三章中介绍）的速度 $\vec{v}_\infty^-$ 和飞出影响球的速度 $\vec{v}_\infty^+$。

已知飞入特定行星的双曲线剩余速度矢量 $\vec{v}_\infty^-$ 和飞越双曲线近心点高度 r_p，即可求出飞越轨道飞出行星的双曲线剩余速度矢量 $\vec{v}_\infty^+$。具体推导过程见第三章。

在行星质心惯性坐标系中，当航天器在脱离引力影响球后，其相对太阳的轨道是由其日心速度 $\vec{v}^+$ 决定的。

如图 2 - 9 所示，行星速度为 $\vec{v}_{\mathrm{Pl}}$，$\vec{v}^-$ 和 $\vec{v}^+$ 分别为航天器绕行星借力前后的日心速度矢量，$\vec{v}_\infty^-$ 和 $\vec{v}_\infty^+$ 为绕行星借力前后航天器相对行星 Pl 的速度矢量。

由此可以得出，借力飞行变轨后，航天器在日心坐标系中的速度变为：

$$\vec{v}^+ = \vec{v}_{\infty^+} + \vec{v}_{\mathrm{Pl}} \tag{2-41}$$

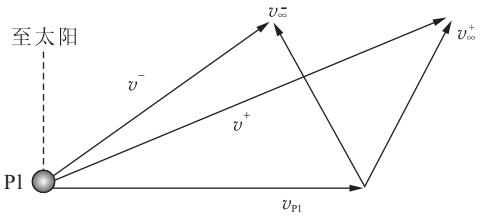


图 2-9 借力飞行日心速度示意图

不难看出，若$|\vec{v}^-|<|\vec{v}^+|$，则航天器飞越行星 Pl 后的日心惯性坐标系中速度增加，反之则减少。事实上，若将航天器和行星 Pl 看作一个整体，根据动量守恒原理，航天器获得了行星 Pl 的部分动量速度增加，而因为行星质量太大，它的速度减少量极小，可以忽略。

第三章　行星际脉冲转移轨道设计

行星际飞行轨道的设计一般分为三段(杨嘉墀,2005),即逃逸轨道段、日心过渡轨道段和遭遇俘获轨道段,其中日心过渡段的设计是轨道优化问题的关键。

在第二章中,介绍了二体和多体问题的运动方程,不论是基于二体还是多体模型,都可以根据给定的初始条件计算任意时刻航天器的理论位置。但在深空探测任务中,必须解决的另一个问题就是交会,即:给定初始时刻 t_0 和该时刻航天器的初始位置,终端时刻 $t_0+\Delta t$ 和该时刻航天器应抵达的位置,来确定在初始位置的速度矢量。

二体模型中,这个问题又称为 Lambert 问题(Bate,*et al*,1990)(还可称高斯问题)。事实上,这一问题也是常微分方程两点边值问题的一种(朱仁璋等,2006)。

简单的近距离交会,可以基于旋转坐标系,使用 CW 方程(又称 Hill 方程)进行求解,但该方法有一定的局限性,不适合用于求解行星际转移轨道。

§3.1　Lambert 问题

行星际日心转移轨道可以认为是点到点的转移,如图 3-1 所示。

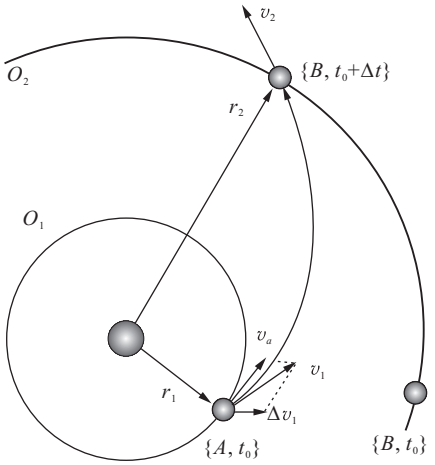


图 3-1　行星际交会轨道示意图

t_0 时刻，行星 A 在 $\vec{r}_1$ 处，t_0 时刻发射航天器，逆时针方向绕日心椭圆轨道运行，经过 Δt 后，与行星 B 在 $\vec{r}_2$ 处交会。对这一类给定 $\vec{r}_1$、$\vec{r}_2$ 以及从 $\vec{r}_1$ 到 $\vec{r}_2$ 的飞行时间 Δt 和运动方向，求解初始速度矢量 $\vec{v}_1$ 和终端速度 $\vec{v}_2$ 的问题称之为 Lambert 问题。

Lambert 问题的求解方法有很多(Bate，*et al*，1990)。大部分都需要进行迭代求解，为了说明求解原理，并为后续计算复杂度分析作简单铺垫，下面就 p 迭代方法进行介绍。

由矢量共面的基本特性可知，若 A、B 和 C 为共面矢量，且 A、B 不共线，则 C 可由 A、B 的线性组合表示。根据二体轨道特性可知，轨道运行于一个平面上，故有矢量关系如下：

$$\begin{aligned}\vec{r}_2 &= f\vec{r}_1 + g\vec{v}_1\\ \vec{v}_2 &= \dot{f}\vec{r}_1 + \dot{g}\vec{v}_1\end{aligned} \tag{3-1}$$

由式(3-1)可得：

$$\vec{v}_1 = (\vec{v}_2 - \dot{f}\vec{r}1)/\dot{g} \tag{3-2}$$

式(3-1)和式(3-2)中，f，g，$\dot{f}$，$\dot{g}$ 满足：

$$\begin{aligned}f &= 1 - \frac{r_2}{p}(1-\cos\theta) = 1 - \frac{a}{r_1}(1-\cos\delta)\\ g &= \frac{r_1 r_2 \sin\theta}{\sqrt{\mu p}} = \Delta t - \sqrt{\frac{a^3}{\mu}}(\delta - \sin\delta)\end{aligned} \tag{3-3}$$

$$\begin{aligned}\dot{f} &= \sqrt{\frac{\mu}{p}}\tan\frac{\theta}{2}\left(\frac{1-\cos\theta}{p} - \frac{1}{r_1} - \frac{1}{r_2}\right) = \frac{-\sqrt{\mu\theta}}{r_1 r_2}\sin\delta\\ \dot{g} &= 1 - \frac{r_1}{p}(1-\cos\theta) = 1 - \frac{a}{r_2}(1-\cos\delta)\end{aligned} \tag{3-4}$$

式中：a 为转移轨道半长轴；

δ 为始末位置矢量 $\vec{r}_1$、$\vec{r}_2$ 之间的夹角；

θ 为始末位置偏近点角角度差；

μ 为中心引力体(太阳)的引力常数。

可见 $\vec{v}_1$、$\vec{v}_2$ 可由 f、g、$\dot{f}$、$\dot{g}$ 以及 2 个已知的位置矢量 $\vec{r}_1$、$\vec{r}_2$ 求得，进而 Lambert 问题可以转化为计算 4 个标量。式(3-3)和式(3-4)中共有 7 个变量，分别是：r_1、r_2、θ、Δt、p、a 和 δ，其中 p、a 和 δ 是未知的，p 可由迭代方式求解。

由式(3-3)和式(3-4)可知：

$$p = \frac{r_1 r_2(1-\cos\theta)}{r_1 + r_2 - 2\sqrt{r_1 r_2}\cos\frac{\theta}{2}\cos\frac{\delta}{2}} \tag{3-5}$$

定义 k、l、m 这 3 个中间量：$k = r_1 r_2(1-\cos\theta)$，$l = r_1 + r_2$，$m = r_1 r_2(1+\cos\theta)$ 。则轨道半长轴可以表示为：

$$a = \frac{mkp}{(2m - l^2)p^2 + 2klp - k^2} \tag{3-6}$$

由式(3－6)可以看出，只要给定一个 p，就可以解出唯一的 a。由式(3－3)和式(3－4)可确定：

$$\cos\theta = 1 - \frac{r_1}{a}(1 - f) \tag{3-7}$$

$$\sin\theta = \frac{-r_1 r_2 \dot{f}}{\sqrt{\mu a}} \tag{3-8}$$

若 a 为负值：

$$\cosh\theta = 1 - \frac{r_1}{a}(1 - f) \tag{3-9}$$

由式(3－3)可确定时间方程如下：

$$\Delta t = g + \sqrt{\frac{a^3}{\mu}}(\theta - \sin\theta) \tag{3-10}$$

对于双曲线方程，则由：

$$\Delta t = g + \sqrt{\frac{(-a)^3}{\mu}}(\sinh\theta - \theta) \tag{3-11}$$

结合式(3－6)不难看出，Δt 是关于 p 的函数，Bate 等(1990)中提出了给定 r_1、r_2 后，飞行时间 Δt 随 p 的变化图(图 3－2)。

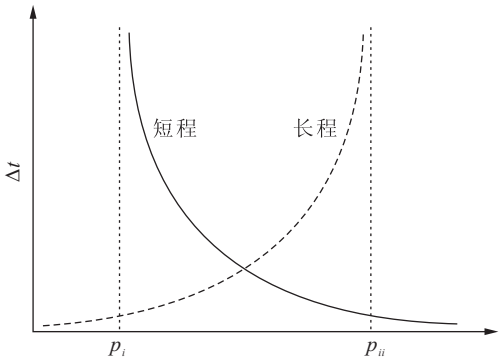


图 3－2　Δt 随 p 的变化典型曲线

图中 p_i 和 p_{ii} 计算公式如下：

$$p_i = \frac{k}{l + \sqrt{2m}} \tag{3-12}$$

$$p_{ii} = \frac{k}{l - \sqrt{2m}} \tag{3-13}$$

可以看出，Δt 与 p 的函数关系单调，可使用二分迭代(石再明等，2006)或线性差值等方

法进行迭代求解。Bate 等(1990)给出了更高效的迭代方法。

总之 Lambert 问题是初轨设计中的一个重要子问题,可形式化描述如下:

$$(\vec{v}_1,\vec{v}_2) = \text{Lambert}(t_0,\vec{r}_1,t_0+\Delta t,\vec{r}_2,\text{flag}) \tag{3-14}$$

§3.2 圆锥曲线拼接法

前面介绍了二体轨道模型和基于该模型的 Lambert 问题求解方法,行星际转移轨道的日心转移轨道段的计算也可以基于这些方法。这种模型认为,图 3-1 中所描述的 $\vec{v}_1$ 和 $\vec{v}_2$(航天器在初始和终端时刻的位置)即航天器在脱离引力影响球后(或离行星无穷远处)的日心轨道速度。由于引力影响球半径(或离行星无穷远处)相对整个飞行过程而言可以忽略,整个日心过渡段的初始和终端位置可近似看做初始和终端行星所在的位置。

求解 t_0 时刻从某行星 A 出发,经过 Δt 天之后与行星 B 交会的问题的具体求解流程如下:

(1)通过星历计算,求出行星 t_0 时刻行星 A 在日心黄道坐标系中的位置 $\vec{r}_1$、速度 $\vec{v}_a$ 和 $t_0+\Delta t$ 时刻行星 B 的位置 $\vec{r}_2$、速度 $\vec{v}_b$;

(2)根据 Lambert 问题,解出 t_0 和 $t_0+\Delta t$ 时刻航天器的日心轨道速度 $\vec{v}_1$、$\vec{v}_2$;

(3)计算航天器两次变轨的脉冲大小($\Delta\vec{v}$):$\Delta\vec{v}_1=\vec{v}_1-\vec{v}_a$ 和 $\Delta\vec{v}_2=\vec{v}_2-\vec{v}_b$;

(4)根据第三步计算得到的 $\Delta\vec{v}$ 设计逃逸和俘获段轨道。

若航天器飞行过程中使用借力飞行变轨,则在上述的过程中需要加入借力飞行轨道段的设计,借力飞行过程近似为瞬间完成,从日心坐标系看,航天器在借力飞行行星质心位置处发生瞬变。

考虑多借力飞行变轨问题:

$$S = \{\text{Pl}_0,\text{Pl}_1,\cdots,\text{Pl}_{n-1}\}$$

$$t = \{t_0,t_1,\cdots,t_{n-1}\}$$

式中:S 为航天器飞行过程中轨道机动所涉及的行星序列(初始位置,借力飞行行星,目标行星);

t 为轨道机动的时刻(使用 MJD2000 描述);

n 为行星个数。

根据这些参数,就可以使用 3.1 节中介绍的 Lambert 问题求出每一段日心过渡轨道。如第 $i(0\leqslant i<n)$ 段日心转移轨道可由 P_i 和 P_{i+1} 在 t_i 和 t_{i+1} 时刻的位置 $\vec{r}_i$ 和 $\vec{r}_{i+1}$,以及转移时长 Δt_i 来求解。

由此不难求出各个 Lambert 转移轨道段的初始速度 $\vec{v}_i^+$ 和终端速度 $\vec{v}_i^-$。下面介绍根据各 Lambert 段的轨道如何确定逃逸段、俘获段和借力飞行段的轨道。在介绍圆锥曲线拼接法前,必须先介绍引力影响球的概念。

3.2.1　引力影响球

将行星际空间按各行星的引力大小划分为各自的作用域，称为该行星的影响球，在影响球内飞行的航天器受到该行星的引力大于其他天体的引力，即可以忽略其他天体对航天器的影响，按前面所述的二体运动方程进行计算，当航天器进入另一个行星的影响球，则以另一行星为中心引力体，构成新的二体问题。影响球的划分对于行星际飞行阶段很重要。

为估计行星引力影响球的半径，Laplace 推导的计算公式，在太阳系中，各行星的引力影响球半径 R 为（耿长福，2006）：

$$R = \rho\left(\frac{m_p}{m_s}\right)^{\frac{2}{5}} \tag{3-15}$$

式中：m_p 为行星的质量；

m_s 为太阳的质量；

ρ 为行星与太阳之间的距离。

航天器接近某行星时，若该行星与航天器的距离小于该影响球的半径 R，即航天器已飞入影响球内，则认为该行星的作用力大于任何其他天体的作用力，可以忽略其他天体的影响。相反，当航天器远离该行星后，若它们之间的距离大于影响球半径 R，则可忽略该行星引力对航天器的影响。

太阳系内各行星的影响球半径见附录 B。

3.2.2　逃逸段

假设首段日心转移轨道已经计算出来：已通过求解 Lambert 问题确定了，航天器在 t_0 时刻从 P_0（地球）出发，经过 Δt_1 到达 p_1 的轨道，并已知轨道初始速度为 $\vec{v}_0^+$。如图 3－3 所示。

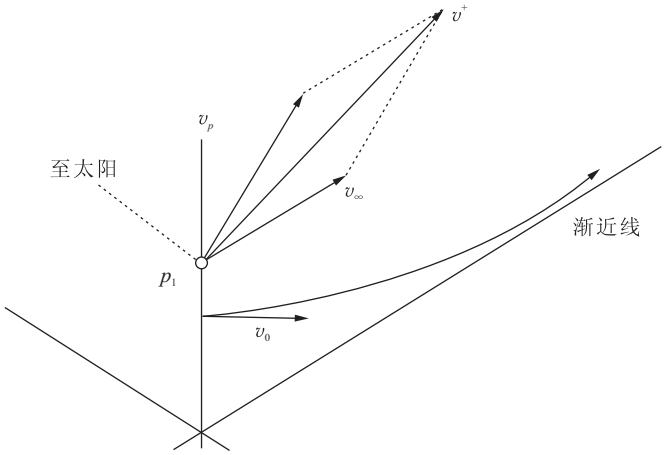


图 3－3　逃逸轨道示意图

在图 3－3 中，$\vec{v}_p$ 为日心坐标系中地球的公转速度；$\vec{v}_0$ 为地心坐标系中双曲线逃逸轨道近地点速度；$\vec{v}_\infty$ 为航天器逃逸双曲线剩余速度（地心坐标系）；$\vec{v}^+$ 为航天器在日心坐标系中的速度（Lambert 转移轨道段的初始速度）。

其中：

$$\vec{v}_\infty \approx \Delta\vec{v} = \vec{v}^+ - \vec{v}_p \tag{3-16}$$

式（3－16）实际上将日心坐标系中的 $\Delta\vec{v}$ 转换到了地心坐标系中的 $\vec{v}_\infty$。由于圆锥曲线拼接法是简化模型，因此 $\Delta\vec{v}$ 和 $\vec{v}_\infty$ 仅在数值上近似（耿长福，2006），而实际上我们关心的也只是 $|\vec{v}_\infty|$ 的值。

航天器在首次变轨前，通常在地球停泊轨道上绕飞，假设停泊轨道近地点高度（距离地心的距离）为 h，轨道偏心率为 e，且航天器在近地点处施加脉冲进入逃逸双曲线轨道，（停泊轨道与双曲线逃逸轨道的近地点重合）。那么不难计算近地点需要施加脉冲得到的速度变量 $|\Delta\vec{v}_0|$。

圆锥曲线方程式（2－20）和轨道半长轴 a 与通径 p 的关系：

$$p = a(1-e^2) \tag{3-17}$$

结合公式（2－26）若已知近地点高度 r_p 和轨道偏心率 e，不难得出：

椭圆轨道近地速度 $\vec{v}_0$ 的大小为：

$$|\vec{v}'_0| = \sqrt{\frac{\mu(1+e)}{r}} \tag{3-18}$$

由能量公式可得双曲线轨道近地速度 $\vec{v}_0$ 的大小为：

$$|\vec{v}_0| = \sqrt{\frac{2\mu}{r} + |\vec{v}_\infty|^2} \tag{3-19}$$

最后得到近地点需要施加脉冲以改变速度的大小为：

$$|\Delta\vec{v}| = |\vec{v}_0| - |\vec{v}'_0| = \sqrt{\frac{2\mu}{r} + |\vec{v}_\infty|^2} - \sqrt{\frac{\mu(1+e)}{r}} \tag{3-20}$$

3.2.3 俘获段

俘获段与逃逸段的过程刚好相反，逃逸段是由停泊轨道近地点施加脉冲以摆脱地球引力进入逃逸双曲线，而俘获段则是通过已知的双曲线剩余速度大小和目标轨道的参数（近心点高度和偏心率）来计算需要多大的 Δv 来进入目标轨道（表 3－1）。计算公式与式（3－20）一致。

值得注意的是，对于逃逸轨道，人们通常只关心需要达到的双曲线剩余速度的大小 $|\Delta\vec{v}_\infty|$，该值通常与逃逸所需要的燃料消耗成正比，有时也用 $C3$ 来衡量（Aerospace，2001）。

$$C3 = |\Delta\vec{v}_\infty|^2 \tag{3-21}$$

但对于俘获轨道，通常根据行星探测目的不同，接近行星后需要的机动量也不一样。

表 3－1　目标行星附近的脉冲机动

探测目的	附加参数	脉冲机动大小(Δv)
飞越	无	0
绕飞	目标轨道近心点高度 r_p，目标轨道偏心率 e	式(3－20)
着陆	无	式(3－16)

3.2.4　深空机动(DSM)

深空机动又称深空点火，属于脉冲变轨类型。前面介绍的脉冲机动，通常是指在行星附近进行的机动，如从停泊轨道进入转移轨道，深空机动则被认为是非轨道校正性质的大推力脉冲转移，通常发生在离行星较远处。例如许多文献中提到的多脉冲 Lambert 交会问题(齐映红等，2006)，其中间过程中所加的脉冲就可以认为是深空机动。

DSM 分为位置模型和速度模型两种。位置模型是指在惯性坐标系中指定 DSM 的位置矢量 P_{dsm} 和施加 DSM 脉冲的时刻，将整个过程分为前后两段 Lambert 问题；而在某些轨道优化问题中，人们通常在给定位置，为航天器尝试不同的初始速度，在飞行一段时间后施加 DSM 脉冲，通过求解 Lambert 问题来与目标行星交会，故称这种 DSM 为速度模型。

许多文献也证明，从 Δv 极小的角度出发，三脉冲和四脉冲 Lambert 交会，通常要比双脉冲 Lambert 交会模型更优(齐映红等，2006)。但向开恒等(1999)也证明，在固定转移时间的前提下，增加脉冲次数并不能显著减少燃料消耗。因此，脉冲次数并不是越多越好。

如图 3－4 所示，航天器初始轨道与黄道面重合，逆时针方向飞行，交会点在黄道面上的投影与分离点共线，太阳与交会点的连线 OB 与黄道面有一个很小的夹角 α，矢量 OA 与 OB 构成的平面与黄道面垂直，由于转移轨道平面由转移轨道的始末位置决定，因此图中的 Lambert 转移轨道必与黄道面垂直。若 α 角为 0，则转移轨道与具有“最省能量”的霍曼变轨类似，转移轨道平面与初始轨道面重合，转移轨道初始速度与原轨道速度共线，需要较小的 Δv 进行变轨。而图 3－4 中的转移轨道初始速度则与分离前的原轨道速度垂直，这将导致一个更大的 Δv 才能完成这样的变轨。

实际上，行星绕太阳公转的轨道面各有差异，这也导致了依照非共面星历模型计算的行星际最优双脉冲转移轨道不可能是分离点与交会点接近 180°的轨道。

耿长福(2006)、Bate 等(1990)介绍了 Fimple 提出的一种飞行程序：先将航天器发射到一个位于黄道面内的转移轨道上，当航天器离交会点的真近点角还差 90°时(如图 3－5 中的 C 点)，作平面机动变轨，这样的旋转为最小。但文中并未给出平面机动点的求解方法(Ox 轴上有无穷多个点符合要求)，因此引入位置模型 DSM。

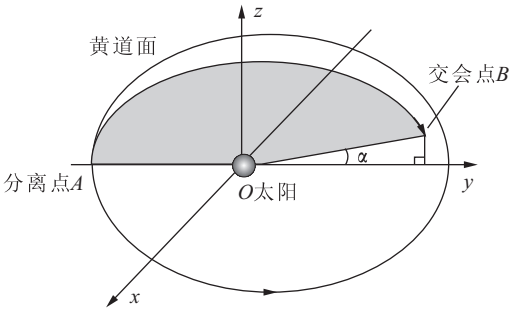


图 3－4　非共面转移轨道示意图

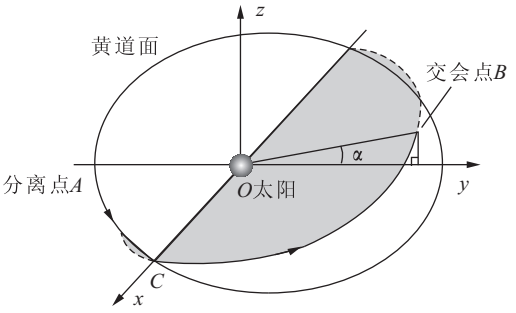


图 3－5　Fimple 最优轨道平面点

位置模型 DSM 通过指定深空机动的位置，来确定转移轨道。实际上可以认为，位置模型 DSM 在整个过程的 Δt 不变的前提下，将一条 Lambert 转移轨道分成了两段 Lambert 转移轨道。

§3.3　借力飞行轨道设计

由于深空探测任务的目标星体距离地球可能很远，如果仅仅使用脉冲变轨，采用相对目标星的直接转移轨道，那么所要求的发射能量会很大，目前的运载火箭可能无法满足其要求。为了节省发射能量，通常先用较小的速度飞行，然后在航行过程中借助行星的引力来加速或改变探测器的飞行方向，从而最终飞向目标星体。这种借助行星引力支援的飞行，即引力辅助变轨，通常称为“借力飞行”。这就是说，在行星际轨道设计中可以利用行星的引力作用改变航天器的运动速度的大小和方向，从而可以在没有任何能量消耗的情况下对航天器加速，从而完成既定的行星际轨道设计。

借力飞行轨道不同于直接转移轨道，因此也需要采用不同的方法来对其进行分析设计。借力飞行技术目前广泛应用于行星际轨道设计中，本节将主要讨论借力飞行的轨道设计。下面详细分析借力飞行技术。

3.3.1　弹弓效应

为了更好地理解借力飞行过程中，航天器相对于太阳的速度（大小和方向），即日心速度发生变化的原因，首先来分析一下物理中的“弹弓效应”现象。

如图 3－6 所示，假设整个过程为弹性碰撞，且小球（图 3－6 中浅色球）质量远小于大球（图 3－6 中深色球）质量。地面上有一静止的大球，小球从某一高度 h 落下，以速度 $\vec{v}$ 与大球发生碰撞，依据动能和动量守恒定律，小球反弹的速度为 $-\vec{v}$，反弹高度为 h。

如果在图 3－6 所示的过程中，大球与小球从某一高度 h 处落下时，如图 3－7(a)所示，

认为下落过程中两球紧挨而不发生接触。以地面作为参考，当大球落地时速度为 $\vec{u}$，大球与地面发生弹性碰撞，依据动能和动量守恒定律，大球反弹的速度为 $-\vec{u}$，而此时小球以速度 $\vec{v}$ 下落。

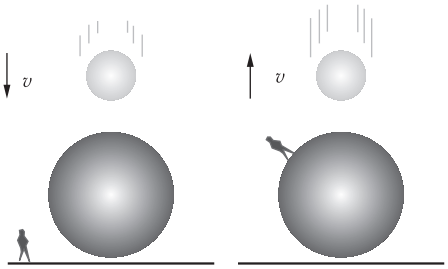


图 3－6　弹弓效应示意图(大球静止)

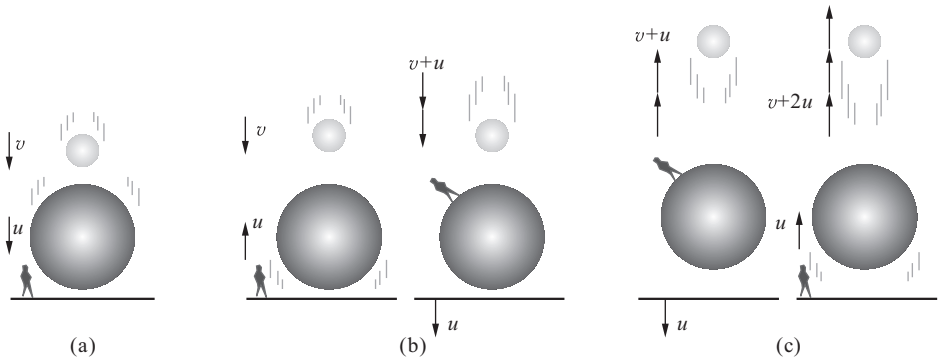


图 3－7　弹弓效应示意图(大球运动)

此时，如图 3－7(b)所示，以地面作为参考，大球速度 $-\vec{u}$，小球速度 $\vec{v}$；若以大球作为参考，则地面的速度为 $\vec{u}$，而小球的速度为 $\overrightarrow{v+u}$。

随后，小球与大球发生碰撞，如图 3－7(c)所示，由于小球质量远小于大球质量，且碰撞为弹性碰撞。以大球作为参考，依据动能和动量守恒定律，地面的速度为 $\vec{u}$，而小球的反弹速度为 $-(\overrightarrow{v+u})$ 向上。而以地面作为参考时，大球的速度为 $-\vec{u}$ 向上，而小球的反弹速度为 $-(\overrightarrow{v+2u})$ 向上。

从分析中可以看到，图 3－6 和图 3－7 之间的唯一区别在于大球的静止与运动。正是因为大球的运动，使得小球的反弹速度不同。大球静止时的反弹速度大小为 v，而大球运动(速度大小为 u)时小球的反弹速度大小为 $v+2u$，相当于得到了两倍的大球速度。

理论上，以自由落体运动分析，如图 3－8 所示，从高度 h 处下落时，小球的反弹高度可

以达到 $9h$(朱仁璋等,2006;吴涛等,2000)。

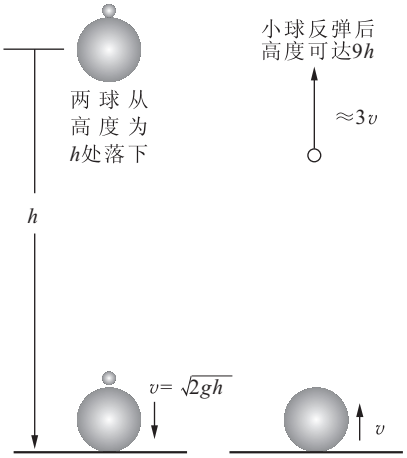


图 3-8 弹弓效应示意图

3.3.2 借力飞行过程分析

借力飞行是一种经常用来减少航天任务中燃料消耗的方法,因为它可以不需要外力改变航天器的速度(大小和方向),借力飞行的过程可以借助于物理学上的“弹弓效应”来进行对比分析。

航天器在日心轨道飞行的过程中,靠近某一天体(一般是行星)时,如图 3-9 所示,当进入该行星的影响球内后,将主要受到该天体的引力作用。在该引力影响球内,构成了以行星引力为中心力场的二体问题,且航天器相对于行星的轨道为双曲线轨道。

航天器以双曲线轨道飞出引力影响球之后,将重新以日心轨道飞行。航天器相对于行星的速度,在进入和飞出引力影响球时,大小不变,只是方向发生了偏转。但是航天器相对于太阳的速度,因为借力天体存在,类似于“弹弓效应”的分析方法,航天器进入和飞出引力影响球时,相对于太阳的速度的大小和方向都发生改变,从而导致其脱离该天体影响球后,日心轨道发生改变,达到变轨的目的。

此过程可与 3.3.1 节讨论的内容可以类比:太阳⇔地面;借力行星⇔大球;航天器⇔小球。类似于“弹弓效应”,我们考虑一种极限情况,如图 3-10 所示,假设航天器以 180°绕行星借力飞行,即类似于小球与大球碰撞的情况。因此借力飞行之后,理论上航天器的速度大小可达到 $v+2u$。

接下来,我们从数学角度分析该过程。假设航天器的质量为 m,借力星体的质量为 M,借力飞行之前,航天器的速度为$\overrightarrow{v_1}$,借力星体的速度为$\overrightarrow{u_1}$,借力飞行之后,它们的速度分别为

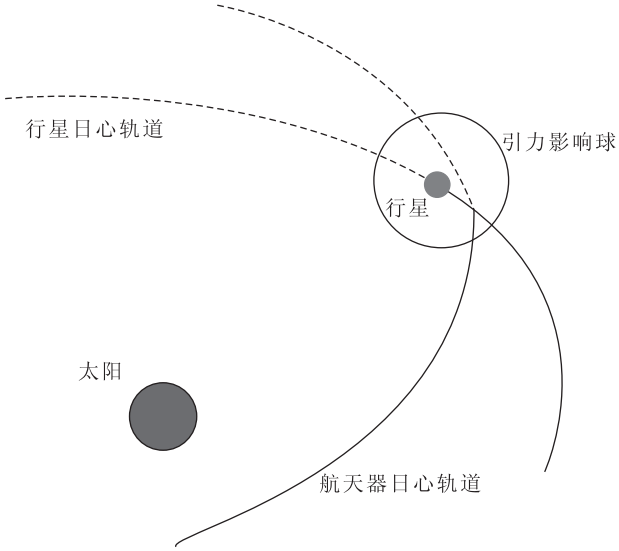


图 3－9　借力飞行过程示意图

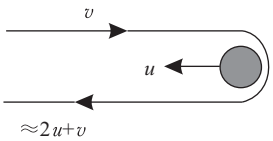


图 3－10　借力飞行的一种极限情形

$\vec{v_2}$和$\vec{u_2}$。因为整个过程只有万有引力做功，所以依据动能和动量守恒定律分析借力飞行可得：

$$\begin{aligned} Mu_1^2 + mv_1^2 &= Mu_2^2 + mv_2^2 \\ Mu_1 - mv &= Mu_2 + mv_2 \end{aligned} \tag{3-22}$$

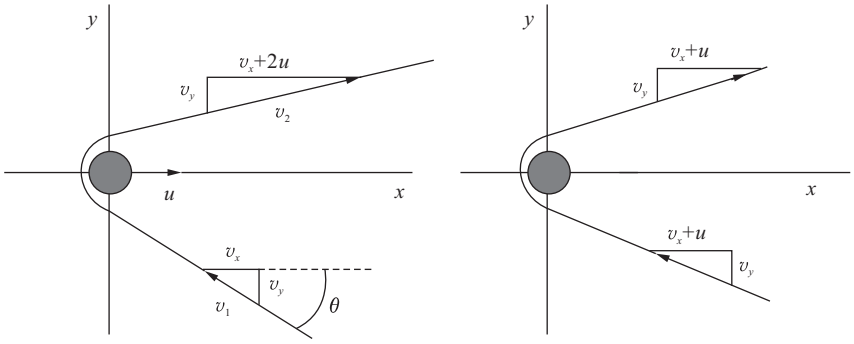
将式(3－22)中 u_2 消去，可以求得 v_2：

$$v_2 = \frac{(1-q)v_1 + 2u_1}{1+q} \tag{3-23}$$

式中：q 为航天器与借力星体的质量之比，即 m/M。

由于 $m \ll M$，因此可以认为 q 接近于 0，所以有 $v_2 = v_1 + 2u_1$。

而实际过程中，这种 180°的速度偏转是不可能达到的，但是其基本规律是一样的。考虑更实际的情况，如图 3－11 所示，以借力星体的速度方向为 x 轴，其垂直方向为 y 轴。航天器以日心速度$\vec{v_1}$飞向借力星体，与 x 轴之间的夹角为$\pi-\theta$，如图 3－11(a)。



(a) 太阳参考系　　(b) 行星参考系

图 3-11　借力飞行

在 x 轴与 y 轴方向，航天器的速度分别为：

$$v_{1x} = -v_1\cos\theta \qquad v_{1y} = v_1\sin\theta \tag{3-24}$$

依据式(3-23)可知，借力飞行之后，沿 x 轴和 y 轴方向的速度分别为：

$$v_{2x} = v_1\cos\theta + 2u_1 \qquad v_{2y} = v_1\sin\theta \tag{3-25}$$

所以，借力飞行之后，航天器的速度为：

$$v_2 = (v_1 + 2u_1)\sqrt{1 - \frac{4u_1v_1(1-\cos\theta)}{(v_1+2u_1)^2}} \tag{3-26}$$

为分析速度的变化与角度 θ 之间的关系，我们假设借力飞行之前，$v_1 = u_1$，则式(3-26)可简化为：

$$v_2 = v_1\sqrt{5 + 4\cos\theta} \tag{3-27}$$

从式(3-27)可知：

当 $\theta=0$ 时，即航天器的飞行方向逆着借力星体的速度方向，借力飞行之后速度是借力飞行之前的 3 倍；

当 $\theta=\pi$ 时，即航天器的飞行方向顺着借力星体的速度方向，借力飞行前后速度不发生变化；

当 $\theta=\pi/2$ 时，即航天器的飞行方向垂直于借力星体的速度方向，则在水平方向可以得到的速度为$\sqrt{5}v_1$。

3.3.3　借力飞行轨道设计

在知道了借力飞行的物理原理和过程之后，就可以根据其自身的特点来设计行星际飞行轨道，充分利用借力飞行的变轨作用，来实现任务设计中的轨道要求。

借用引力影响球的概念，一般的行星际轨道可以分为如下几段（马文臻，2006）：

（1）地球逃逸轨道：摆脱地球引力的轨道，在地球影响球内，航天器以相对于地球的双曲线轨道飞行，忽略太阳以及其他行星影响；

（2）日心过渡轨道：探测器以相对太阳的椭圆轨道运行，忽略太阳以外天体对航天器的影响；

（3）行星绕飞轨道：引力影响球内，忽略其他天体，包括太阳的影响，航天器相对借力行星为双曲线轨道；

（4）目标轨道：航天器被行星捕获进入环绕目标星的轨道，或者绕飞行星后进入日心目标轨道。

在每一段中，航天器的轨道均为圆锥曲线的一种，任务设计就是需要把各段圆锥曲线连接起来，确定任务的初轨。这种分段计算，最后将每一段连接起来得到最终的行星际轨道的方法，就是圆锥曲线拼接法。

借力飞行的原理在前面的有关章节中已有介绍，其关键技术之一是给出行星质心惯性坐标系中航天器飞入双曲线的剩余速度 $\vec{v}_\infty^-$ 和双曲线飞越轨道近心点高度 r_p 来确定飞出速度 $\vec{v}_\infty^+$ 的方向（即图 3－12 中的角度 δ）。下面就这一问题作简单推导。

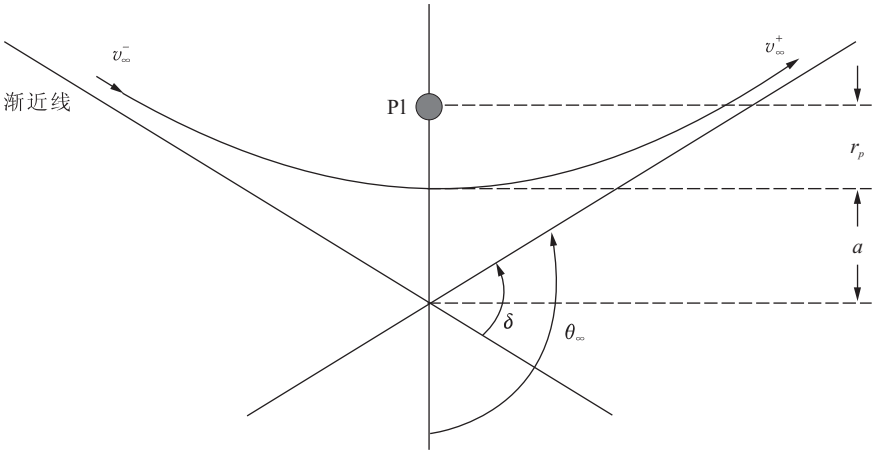


图 3－12　双曲线飞越轨道示意图

双曲线飞越轨道的几何特性如图 3－12 所示，忽略其他星体引力条件下，在行星质心惯性坐标系中，航天器从无穷远处以速度 $\vec{v}_\infty^-$ 接近行星 Pl。图中几个参数意义如下：r_p 为飞越双曲线的近心点高度；μ 为行星引力常数；δ 为 $\vec{v}_\infty$ 的斜偏角，即 $\vec{v}_\infty^-$ 和 $\vec{v}_\infty^+$ 之间的夹角；a 为双曲线飞越轨道半长轴长度；θ_∞ 为双曲线飞越轨道的真近点角。

参数 θ_∞、δ 和 e 可由圆锥曲线方程确定。方程如下：

$$r = \frac{a(1-e^2)}{1+e\cos\theta} \tag{3-28}$$

当 $r \to \infty$时，就有 θ_∞，因此：

$$\cos\theta_\infty = \lim_{r\to\infty}\left\{\frac{1}{e}\left[\frac{a(1-e^2)}{r}-1\right]\right\} = -\frac{1}{e} \tag{3-29}$$

从图 3-12 可得：

$$\theta_\infty = \frac{\pi}{2}+\frac{\delta}{2} \tag{3-30}$$

由式(3-29)和式(3-30)即得到：

$$\sin\frac{\delta}{2} = \frac{1}{e} \tag{3-31}$$

设行星引力常数为 μ，又知，双曲线半长轴为 a 的飞越轨道与剩余速度关系为：

$$v_\infty = \sqrt{\frac{\mu}{a}} \tag{3-32}$$

双曲线轨道偏心率为：

$$e = 1+\frac{r_p}{a} \tag{3-33}$$

由式(3-31)、式(3-32)和式(3-33)不难得出：

$$\delta = 2\arcsin\left[\frac{1}{1+(r_p v_\infty{}^2/\mu)}\right] \tag{3-34}$$

这种借力飞行的过程中不加任何脉冲，航天器绕过行星后可改变其相对太阳的速度矢量，该模型对于给定的飞入双曲线剩余速度 $\vec{v}_\infty^-$，其飞出速度矢量 $\vec{v}_\infty^+$ 也就固定了。

需要说明的是，从前面讨论和分析可以知道，航天器借力飞行之后，速度发生改变，从而实现变轨。但是如果要求变轨之后的轨道就是所需要的轨道，那么就需要考虑行星间的会合周期，从而确定航天器的发射窗口，因此对发射窗口有特定的限制。在实际情况中，我们较少考虑行星会合周期，尤其是多借力飞行时，一般都需要对借力飞行后的轨道进行调整以得到期望的轨道。因此借力飞行可以分为纯借力飞行和带动力借力飞行(Powered Gravity Assist，PGA)。由于纯借力飞行对行星间的会合周期有特定的要求，所以纯借力飞行一般在实际的工程应用中难于实现。

下面讨论带动力借力飞行轨道设计，而对于施加动力的时刻，Prado 等人对此进行了研究，并指出在借力飞行之后，对航天器实施深空机动以调整轨道，效果要优于借力飞行之前和借力飞行过程中实施的动力(Tamás，*et al*，2007)。

若 PGA 变轨前后各为一段 Lambert 日心转移轨道，结合式(3-16)不难得出进入借力行星前后的双曲线剩余速度矢量 $\vec{v}_\infty^-$ 和 $\vec{v}_\infty^+$。则可根据式(3-20)计算出两条双曲线轨道近心点速度大小，可得近心点位置脉冲大小公式如下：

$$|\Delta\vec{v}| = \left|\sqrt{\frac{2\mu}{r_p}+|\vec{v}_\infty^+|^2}-\sqrt{\frac{2\mu}{r_p}+|\vec{v}_\infty^-|^2}\right| \tag{3-35}$$

即近心点处施加的脉冲大小$|\Delta\vec{v}|$是关于近心点高度 r_p 的函数，如图 3－13 所示。

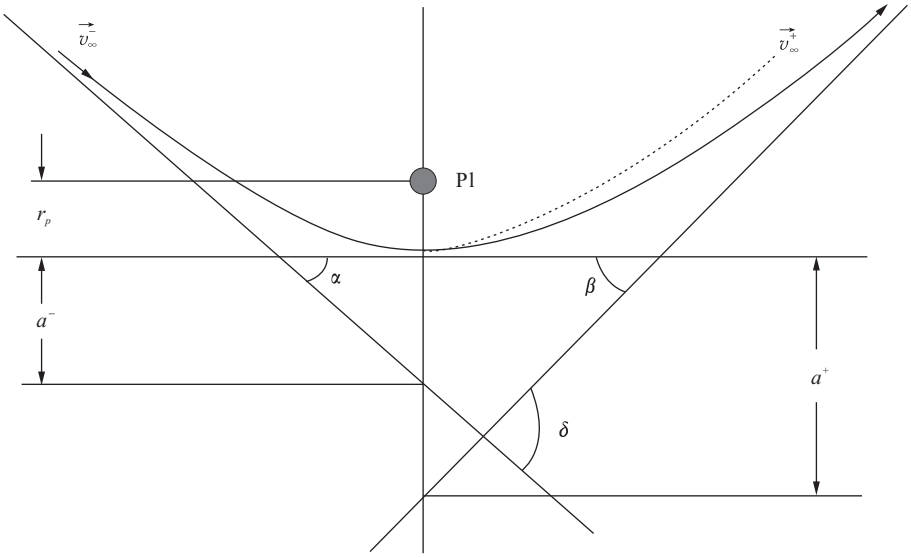


图 3－13　带机动的借力飞行

其中 a^- 和 a^+ 为飞入和飞出双曲线的半长轴，可由下式得出：

$$a = \frac{\mu}{v_\infty^2} \tag{3-36}$$

式中：μ 为行星引力常数。

那么如何确定一个适当的近心点高度 r_p，使得飞出速度矢量在方向上符合要求呢？

设 δ 为已知飞入和飞出双曲线剩余速度 $\vec{v}_\infty^-$ 和 $\vec{v}_\infty^+$ 的夹角，α 和 β 分别为 $\vec{v}_\infty^-$ 和 $\vec{v}_\infty^+$ 与近心点速度 $\vec{v}_0$ 的夹角，由双曲线性质可以得出：

$$\begin{aligned}\alpha &= \arcsin[a^- /(a^- + rp)]\\ \beta &= \arcsin[a^+ /(a^+ + rp)]\end{aligned} \tag{3-37}$$

引入函数 f：

$$\begin{aligned}f &= \alpha+\beta-\delta\\ &= \arcsin[a^- /(a^- + rp)]+\arcsin[a^+ /(a^+ + rp)]-\delta\end{aligned} \tag{3-38}$$

已知矢量 $\vec{v}_\infty^-$ 和 $\vec{v}_\infty^+$，不难得出它们的夹角 δ(即双曲渐近线夹角)，并且由三角形性质有：

$$\delta = \alpha+\beta \tag{3-39}$$

即需要寻找一个适当的 r_p，使得式(3－38)为 0。对 f 求导有：

$$f' = -\frac{a^-}{(a^- + r_p)\sqrt{r_p(r_p + 2a^-)}} - \frac{a^+}{(a^+ + r_p)\sqrt{r_p(r + 2a^+)}} \tag{3-40}$$

根据式(3-40)即可通过牛顿迭代的方式求得满足式(3-39)的 r_p 。若方程无解,则该方案不可行。

图 3-14 是采用基于二体模型的圆锥曲线法求解“地球-金星-火星”借力飞行问题的仿真示意图,即绕金星一次借力飞行,代码使用 VC++编写,三维场景使用 OpenGL 绘制。

图 3-14 仅仅只是一个仿真,左图为完整轨道,右图为航天器飞行轨迹。其中假定航天器从地球发射的时刻是以 MJD2000 为起始时刻的第 80 天,从地球到金星的转移时间为 80 天,而金星到火星的转移时间为 200 天,图中显示了该借力飞行变轨示意图。

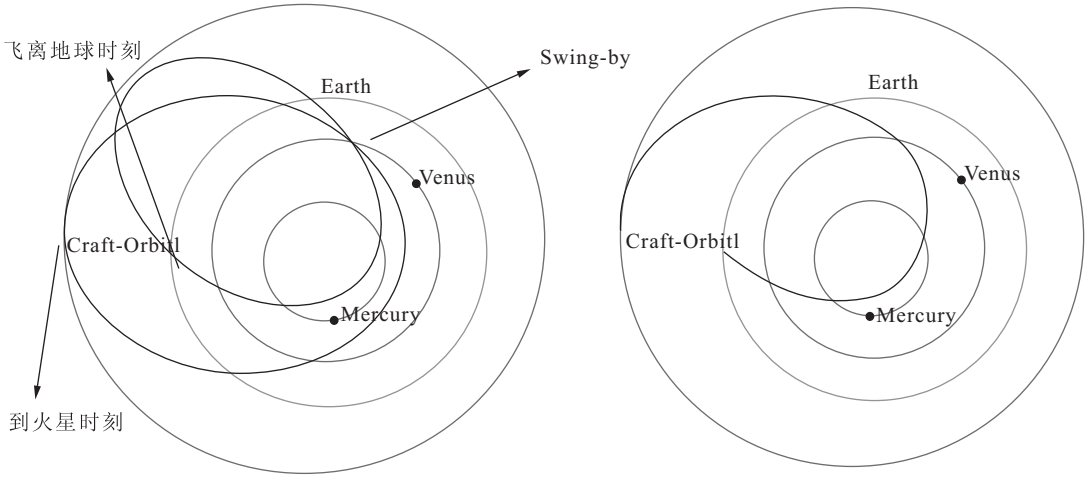


图 3-14 借力飞行仿真示意图

§3.4 B 平面法

圆锥曲线拼接法仅计算出各行星质心轨道上的脉冲大小 $|\Delta\vec{v}|$,而没有其方向信息,从图 3-11 和图 3-12 可以看出,仅给出逃逸双曲线剩余速度矢量($\vec{v}_\infty^-$ 或 $\vec{v}_\infty^+$) 和双曲线近心点高度 r_p 是无法确定轨道平面在三维空间中的方向的。

事实上,这些参数对应着无数条双曲线轨道,以地球逃逸轨道为例,若逃逸轨道近地点高度确定,那么所有满足给定双曲线剩余速度 $\vec{v}_\infty^+$ 的轨线的集合呈现出如图 3-15 所示的一个包络效果。

这些逃逸双曲线轨道有着相同的双曲线剩余速度,它们的近心点初始速度矢量通常作为精确轨道确定时(如微分校正等方法)的初值。而工程上则通常通过轨道倾角的约束(如地球自转和太阳光照等因素)来确定具体的逃逸轨道倾角参数,一旦倾角和发射方向确定,

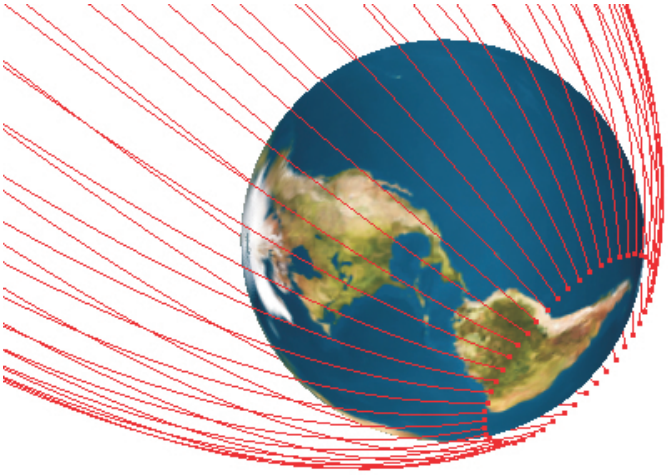


图 3－15　逃逸地球的双曲线轨道示意图

轨道就具有唯一性了。

而对于借力飞行的双曲线轨道(不管是带机动的还是无机动的借力飞行),通常我们需要在初轨设计阶段就确定借力后的剩余速度的方向。而给定的飞入剩余速度,对应这无数条不同方向的飞越双曲线(与图 3－16 中的飞越双曲线绕 $\hat{S}$ 轴旋转形成的包络效果类似)。

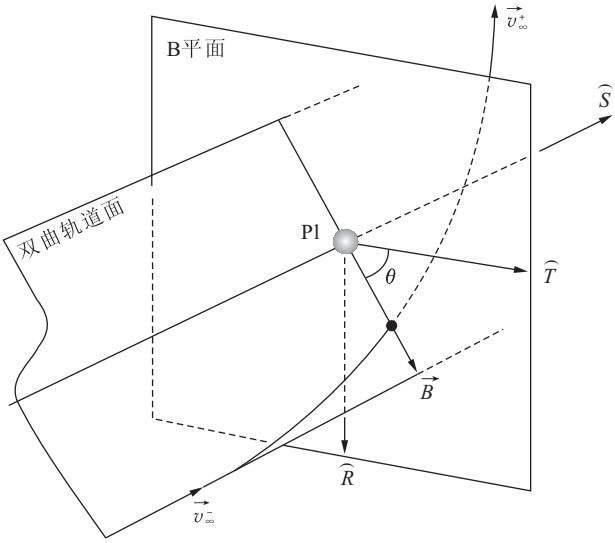


图 3－16　B 平面示意图

为了在轨道设计中参数化行星中心坐标系中的近心点速度和日心坐标系中的行星运行速度的关系，需引入 B 平面参数(George，2005)。

Kizner. W 在 20 世纪 60 年代初发现建立在目标天体 B 平面上的参数与航天器飞行轨道状态参数之间存在很好的线性关系。大量的理论分析和工程实践表明，深空探测航天器的飞行轨道可以用 B 平面参数来描述目标散布，并且用迭代方法修正 B 平面参数的误差有较好的收敛性，因此 B 平面参数广泛应用于深空探测的轨道设计与优化中(谷立祥等，2003)。

B 平面和航天器的轨道平面垂直相交，B 矢量的方向与这两个平面的相交线一致，起始点是目标天体的中心，指向飞入渐近线与 B 平面的交点。B 平面参数建立在以目标天体中心为原点的平面坐标系下。以飞入剩余速度 $\vec{v}_\infty^-$ 的方向为 $\mathring{S}$ 轴，$\mathring{S}$ 与行星 Pl 的日心速度矢量的乘积作为 $\mathring{T}$ 轴，并由右手定则确定 $\mathring{R}$ 轴。

因此，若已知飞入剩余速度矢量 $\vec{v}_\infty^-$ 和近心点高度 r_p，则可通过指定 $\vec{B}$ 矢量与 $\mathring{T}$ 轴的夹角 θ 来唯一地确定飞出剩余速度。方法如下：

已知行星引力常数 μ，飞越双曲线近心点高度 r_p，飞入双曲线剩余速度 $\vec{v}_\infty^-$，

(1)确定 $\mathring{S}$ 轴：$\vec{v}_\infty^-$ 的归一化矢量；

(2)由星历表计算得到当前行星的日心速度矢量 $\vec{v}_{\mathrm{Pl}}$，则 $\mathring{T}=\mathrm{unify}(\vec{v}_{\mathrm{Pl}}\times\vec{v}_\infty^-)$，其中 unify 为将矢量归一化的操作；

(3)将 $\mathring{T}$ 轴绕 $\mathring{S}$ 轴旋转 θ 角，并根据式(3-33)和式(3-36)，将该矢量缩放至飞越双曲线短半轴长度 b，得出矢量 $\vec{B}$；

(4)计算轨道平面法矢量 $\vec{n}$，$\vec{n}=\mathrm{unify}(\vec{B}\times\mathring{S})$；

(5)由式(3-34)计算 $\vec{v}_\infty^-$ 与 $\vec{v}_\infty^+$ 的夹角 δ；

(6)将 $\vec{v}_\infty^-$ 绕法矢量 $\vec{n}$ 旋转 δ，即得到 $\vec{v}_\infty^+$。

将矢量绕任意轴旋转的方法见附录 A。行星际探测任务设计中，在到达目标行星后，若目标为绕飞或飞越，则 B 平面方法同样适用于目标行星俘获段的设计。

行星际轨道探测中，通常需要使用 B 平面方法来进行精确轨道设计。本书讨论初轨设计，仅在圆锥曲线拼接法中使用该方法来参数化借力飞行的双曲线飞越轨道和俘获轨道的轨道平面方向。

§3.5 逐次逼近打靶法求解多体问题下行星际转移轨道

在考虑多体问题的情况下，根据牛顿第二定律，在惯性坐标系下航天器的动力学方程可以写成(2-32)的形式，设第 i 个天体表示航天器，则转移轨道的动力学方程可以写成以下形式：

$$\begin{cases}\ddot{\vec{r}} = f(\vec{r},\dot{\vec{r}},t) \\ \vec{r}(t_0) = \vec{r_0},\vec{r}(t) = \vec{r_t}\end{cases} \tag{3-41}$$

式中：$\vec{r_0}$ 是 t_0 时刻的位置；

$\vec{r_t}$ 是 t 时刻卫星的位置；

$\dot{\vec{r}}_t$ 是 t 时刻卫星的速度矢量；

$\ddot{\vec{r}}_t$ 是 t 时刻卫星的加速度矢量；

t_0、t 分别表示发射时刻和到达时刻的时间。

式(3-41)是一个非线性的常微分边值问题。

3.5.1　打靶法

打靶法的基本思想是对于一个两点边值问题，将其转换为初值问题。在转移轨道设计中，要求解的是航天器发射时的初始速度，由于知道了目标位置，可以估计一个初始速度后进行积分，分析积分终点与目标点的偏移量来计算初始值修正向量，试验修正初值反复迭代，直到收敛。对于非线性边值问题式(3-41)，其初值问题：

$$\begin{cases}\ddot{\vec{r}} = f(\vec{r},\dot{\vec{r}},t) \\ \vec{r}(t_0) = \vec{r_0},\dot{\vec{r}}(t_0) = \vec{v_0}\end{cases} \tag{3-42}$$

的解记为

$$\vec{r}(t) = \vec{\omega}(t,\vec{v_0}) \tag{3-43}$$

如果 $\vec{v_0}$ 满足 $\vec{\omega}(t,\vec{v_0})=\vec{r_t}$，则 $\vec{r}(t)=\vec{\omega}(t,\vec{v_0})$ 就是边值问题式(3-41)的解。记

$$\vec{F}(\vec{v_0}) = \vec{\omega}(t,\vec{v_0}) - \vec{r_t} \tag{3-44}$$

则求解线性边值问题式(3-41)的打靶法即用某种迭代法求函数方程 $F(\vec{v_0}) = 0$ 的根 $\vec{v_0}^*$，在迭代过程中不断地调整 $\vec{v_0}$ 使得初值问题(3-42)的解满足边值问题式(3-41)在 t 时刻的边界条件(刘林等，2005)。

3.5.2　逐次逼近打靶法

逐次逼近打靶法是在打靶法的基础上进一步细分得来的，其基本思想是：如果当前要到达目标点为 $\boldsymbol{R}$，假设当前求出的一个假设值 $\boldsymbol{v}_0^*$ 通过积分能够到达一个终点 $\boldsymbol{R}^*$，则将区间 $[\boldsymbol{R}^*,\boldsymbol{R}]$ 划分为 n 个子区间：$[\boldsymbol{R}^*,\boldsymbol{R}_1]$，$[\boldsymbol{R}_1,\boldsymbol{R}_2]$，…，$[\boldsymbol{R}_{n-2},\boldsymbol{R}_{n-1}]$，$[\boldsymbol{R}_{n-1},\boldsymbol{R}]$，将打靶目标分解为 n 个阶段，即将打靶目标依次确定为：$\boldsymbol{R}_1,\boldsymbol{R}_2,\cdots,\boldsymbol{R}_{n-2},\boldsymbol{R}_{n-1}$，最终到达目标点 $\boldsymbol{R}$。将这种改进的打靶法称为逐次逼近打靶法，如图 3-17 所示。

其算法流程如下：

(1)计算发射行星(如地球)和目标行星(如火星)的星历，求出该发射行星在发射时刻的

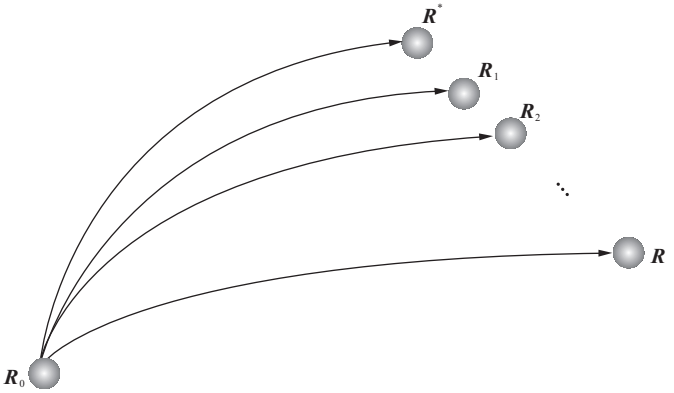


图 3-17 逐次逼近打靶法的示意图

位置 $\boldsymbol{R}_0$ 和目标行星终点时刻的位置 $\boldsymbol{R}$；

(2)通过 $\boldsymbol{R}_0$ 和 $\boldsymbol{R}$ 求解航天器在二体问题情况下转移轨道的初始速度 $\boldsymbol{v}_0^*$；

(3)$\boldsymbol{v}_0^*$ 通过积分能够到达一个终点 $\boldsymbol{R}^*$；

(4)将区间$[\boldsymbol{R}^*,\boldsymbol{R}]$划分为 n 个子区间：$[\boldsymbol{R}^*,\boldsymbol{R}_1]$，$[\boldsymbol{R}_1,\boldsymbol{R}_2]$，…，$[\boldsymbol{R}_{n-2},\boldsymbol{R}_{n-1}]$，$[\boldsymbol{R}_{n-1},\boldsymbol{R}]$；

(5)初始位置为 $\boldsymbol{R}^*$；

for $i=1:n$

a) 初始速度 $\boldsymbol{v}_i^*=\boldsymbol{v}_{i-1}^*$，目标位置为 $\boldsymbol{R}_i$；

b) 由 $\boldsymbol{R}_{i-1}$ 和 $\boldsymbol{v}_i^*$ 为初始状态 $\boldsymbol{X}_i^*$，对 $\boldsymbol{X}_i^*$ 选取状态增量做 6 次积分，求解状态转移矩阵 φ_{12}；

c) 采用初等变换对状态转移矩阵求逆 φ_{12}^{-1}；

d) 以 $\boldsymbol{X}_i^*$ 为初始状态，用 Rung-Kutta 从 t_0 积分到终点时刻 t，得到终点位置 $\boldsymbol{R}'_i$；

e) 判断 $\boldsymbol{R}'_i$ 与目标点位置 $\boldsymbol{R}_i$ 足够接近，即 $|\boldsymbol{R}'_i-\boldsymbol{R}_i|<\varepsilon$，则结束本次循环；否则执行 $\Delta\boldsymbol{R}_i=\boldsymbol{R}'_i-\boldsymbol{R}_i$，$\boldsymbol{v}_i^{*(k+1)}=\boldsymbol{v}_i^{*(k)}+\varphi_{12}^{-1}\Delta\boldsymbol{R}_i$，转向 b)；

end

(6)最后一次迭代的结果即为所求结果。

3.5.3 实验结果

以地球到火星转移轨道为例，基于双脉冲变轨，航天器的初始位置位于距离地表某一高度的停泊轨道，在初始时刻给一个脉冲使得航天器达到发射速度，到达目标点时对航天器施加另外一个脉冲，使航天器进入俘获轨道。采用的坐标系是日心黄道坐标系，时间系统是

MJD2000，实验时将逐次逼近的打靶法与普通的打靶法进行了比较（表 3－2～表 3－5）。

任务目标：到达日期 2010 年 9 月 2 日（MJD2000：3897.237），选择不同的近火点高度得到实验结果。

约束条件：设转移轨道从近地点起始，转移初速度瞬时获得（即冲量情况），发射时间为 2009 年 10 月 13 日（MJD2000：3573.19），停泊轨道高度 200 km。

表 3－2　距离火星表面 96 907.73km（$n=3$）

算　法	结果误差	迭代次数
打靶法	0.159 921	19
逐次打靶	0.502 850	20

表 3－3　距离火星表面 6 654.86km（$n=10$）

算　法	结果误差	迭代次数
打靶法	0.878 867	95
逐次打靶	0.772 722	62

表 3－4　距离火星表面 1 988.16km（$n=3$）

算　法	结果误差	迭代次数
打靶法	——	＞200
逐次打靶	0.965 247	46

表 3－5　距离火星表面 243.32km（$n=3$）

算　法	结果误差	迭代次数
打靶法	——	＞200
逐次打靶	0.505 787	27

实验结果表明，在目标点距离火星较远的情况下，逐次逼近打靶法和普通打靶法的差别较小。当目标点距离火星表面较近的时候，逐次逼近打靶法有更快的收敛速度，而且在有些情况下普通的打靶法无法收敛时，逐次逼近打靶法能够较好地求得结果。

实验表明，采用逐次逼近打靶法的应用范围比普通的打靶法更广，逐次打靶法也存在一个缺点即划分小区间之后在每个区间内进行打靶时都要求一次状态转移矩阵，而普通的打靶法只需要求一次矩阵。但是，在很多情况下普通的打靶法在求解时是非常困难的，而采用逐次打靶法可以很容易的得到结果。

第四章　行星际轨道优化模型

本书中的行星际轨道优化是指，在给定的发射时间范围内，采用特定的借力飞行序列，并在给定的时间范围内到达目标行星，在这样的前提下寻找一个适当的发射时机，使得飞行所需要的燃料和飞行时长尽量地小。可用于行星际轨道优化的算法很多，如传统的单纯形算法、序列二次规划(SQP)算法以及随机优化算法，如演化算法等。

§4.1　优化模型

4.1.1　MPGA＋位置模型 DSM

考虑带 n 次机动的借力飞行 MPGA＋DSM 行星际脉冲转移轨道设计模型：

$$\vec{P} = \{\vec{P}_0, \vec{P}_1, \vec{P}_2, \cdots, \vec{P}_{n-1}\} \tag{4-1}$$

式中：$\vec{P}$ 为脉冲机动位置矢量序列；对于逃逸、借力或俘获，$\vec{P}_i$ 即为行星位置矢量；对于深空机动，$\vec{P}_i$ 为脉冲机动位置矢量。

根据探测目标的不同，还需要额外参数见表 3－1 中所述参数。

决策向量：

$$X = \{t_0, \vec{P}_0, \Delta t_1, \vec{P}_1, \Delta t_2, \vec{P}_2, \cdots, \Delta t_{n-1}, \vec{P}_{n-1}\} \tag{4-2}$$

X 为轨道脉冲机动发生的时刻序列(MJD2000)。如图 4－1 所示。

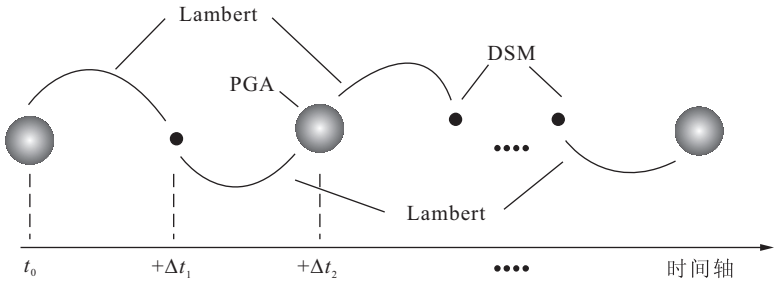


图 4－1　MPGA＋DSM

深空机动的位置矢量通常需要指定(日心黄道坐标系 $O_s x_s y_s z_s$)。而行星位置则来自行星星历表,计算方法如下:

$$\vec{P}_i = \begin{cases} O_s x_s y_s z_s \mathrm{Vector}(x, y, z) \\ \mathrm{eph}(\mathrm{Pl}_i, t_i) \end{cases} \tag{4-3}$$

其中:

$$t_i = t_0 + \sum_{j=1}^{i} \Delta t_j \tag{4-4}$$

$\mathrm{eph}(\mathrm{Pl}_i, t_i)$计算行星 Pl_i 于 t_i 时刻在 $O_s x_s y_s z_s$ 中的位置矢量。星历计算模型将在第九章中展开讨论。

优化目标:

$$\begin{aligned} &\text{minimize}: f_1(X) = \sum_{i=0}^{n-1} \Delta v_i \\ &\text{minimize}: f_2(X) = \sum_{i=1}^{n-1} \Delta t_i \end{aligned} \tag{4-5}$$

约束条件:

$$r_p(X) \geqslant r_{p\min} \tag{4-6}$$

式中:r_p 为借力飞行的飞越双曲线近心点高度;

$r_{p\min}$ 则是高度约束。对于不同的行星,此约束值也不同,最直接的约束就是近心点高度必须比行星半径要大。当然根据实际情况,还可以设置不同的约束值。

MPGA+位置模型 DSM 的优化问题可描述为:指定目标行星和目标轨道参数,给定借力飞行行星序列,在指定的发射和入射窗口内,寻找一个适当的发射时机,并寻找适当的深空机动位置,使得探测任务所需要的能量尽量小,任务时长尽量短。

4.1.2　MGA+速度模型 DSM

与 MPGA+位置模型 DSM 不同的是,MGA+速度模型 DSM 任务中,借力飞行过程中无需施加脉冲,DSM 也不是在指定位置施加,而是通过 DSM 前所在轨道的初始速度和飞行时长来确定施加 DSM 的位置,故称为"速度模型"。

考虑涉及 n 颗行星的行星际探测任务:

$$\mathrm{Pl} = \{\mathrm{Pl}_0, \mathrm{Pl}_1, \cdots, \mathrm{Pl}_{n-1}\} \tag{4-7}$$

Pl 为行星序列,其中 Pl_0 为出发行星(通常为地球),Pl_{n-1}为目标行星,$\mathrm{Pl}_1 \sim \mathrm{Pl}_{n-2}$为借力飞行的行星。根据探测目标的不同,还需要如表 3-1 中所述的附加参数。

决策向量 X 描述如下:

$$X = \{t_0, v_\infty, u, v, \eta_1, \Delta t_1, r_{p2}, \theta_2, \eta_2, \Delta t_2, \cdots, r_{p_{n-1}}, \theta_{n-1}, \eta_{n-1}, \Delta t_{n-1}\} \tag{4-8}$$

式中:n 为行星序列长度;

t_0 为从行星 Pl_0 出发的时刻；

v_∞ 为脱离出发行星的逃逸双曲线剩余速度大小；

u 为 v_∞ 在日心黄道球面坐标系中的黄纬；

v 为 v_∞ 在日心黄道球面坐标系中的黄经；

Δt_i 为第 i 段行星际轨道的时长，$i \in \{1,2,\cdots,n-1\}$；

r_{pi} 为借力飞行或目标行星的双曲线轨道近心点高度；

η_i 中 $i \in \{1,2,\cdots,n-1\}$，表示航天器在第 i 段转移轨道上运行 $\eta_i^* \times \Delta t_i$ 后施加 DSM，后半段通过求解 Lambert 问题实现交会，后半段时长 $(1-\eta_i) \times \Delta t_i$；

θ_i 中 $i \in \{2,3,\cdots,n-1\}$，表示第 i 次借力飞行变轨的 B 平面旋转角。

因此 MGA+速度模型 DSM 任务的优化可描述为：已知航天器飞行过程中涉及的行星序列，找出一个适当的发射时刻、逃逸地球后的双曲线剩余速度矢量、DSM 的时机和各次借力飞行轨道的 B 平面旋转角，以达到降低总能量消耗和减少总飞行时长的目的(图 4-2)。

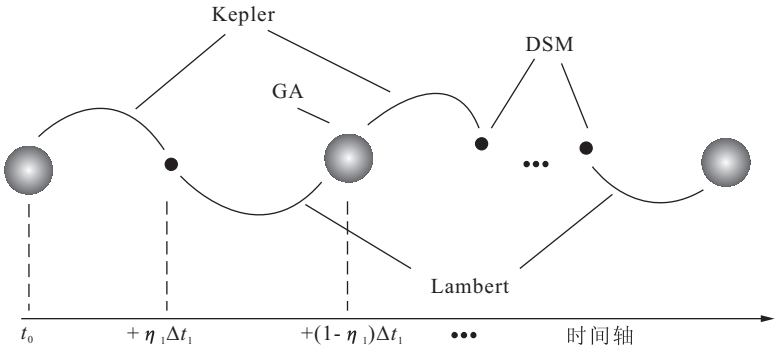


图 4-2 MGA+DSM

优化目标：

$$\begin{aligned} &\text{minimize}: f_1(X) = v_\infty + \sum_{i=0}^{n-1} \Delta v_i \\ &\text{minimize}: f_2(X) = \sum_{i=1}^{n-1} \Delta t_i \end{aligned} \tag{4-9}$$

约束条件同式(4-6)。

§4.2　时空复杂度分析

对优化问题的复杂度分析，有助于很好地理解优化模型的性质和空间剪枝算法的应用，以提高空间搜索效率。本节就一些基本的求解模型和整体优化空间复杂度进行讨论。

4.2.1　Lambert 问题时间复杂度分析

作为轨道设计的基础计算部分，寻优过程中需要进行大量次数的 Lambert 问题求解。这里给出对 p 迭代方法进行的理论分析和统计性实验的结果。

二分 p 迭代的复杂度与迭代区间的大小，以及迭代的收敛精度密切相关。这里对 89 809个算例进行了分析，抽取部分数据如表 4－1 所示。

表 4－1　二分 p 迭代次数统计

迭代精度 m 精确到小数点后 m 位	迭代宽度 W （单位：m）	最坏迭代次数 （理论值）	实际迭代次数 （平均）
0	234 731 727 150.383 24	37.772	25.128
3	234 731 727 150.383 24	47.738	34.092
6	234 744 280 129.057 89	57.704	44.048
9	227 461 332 465.510 13	67.624	52.624

根据算法复杂度理论，为更好地估算连续型区间的优化复杂度，通常将区间内的精度需求和区间宽度结合起来进行计算。结合实验结果不难看出，Lambert 问题求解时间复杂度为：

$$O(\log_2(W \times 10^m)) \tag{4-10}$$

式中：W 为迭代宽度；

m 为迭代精度需求。

另外，在对牛顿迭代次数的统计中，平均迭代次数都可以在 10 次左右。

实验表明，无论是使用二分迭代还是牛顿迭代，就单次求解过程来看，计算复杂度都不高（对数复杂度）。但作为轨道设计的基本算子之一，轨道设计和优化过程中将出现大量次数的求解，因此在寻优过程中，提高求解效率的同时，也应尽量减少求解次数。

4.2.2 双脉冲轨道优化问题复杂度

作为行星际轨道优化问题的最基础问题之一，对于双脉冲转移轨道的优化问题的时空复杂度的研究，对研究 MPGA 和 MGA 等模型的优化问题具有重要价值。

指定出发行星和目标行星的单目标的双脉冲变轨优化问题可描述为：

优化目标：

$$\text{minimize}: f(t_0, \Delta t) = \Delta v_1 + \Delta v_2 \tag{4-11}$$

该问题实际上就是求解一个 Lambert 问题，前面已经讨论过 Lambert 问题求解的复杂度，现在来分析优化问题的空间复杂度。

如图 4-3 中绘出的是地球-火星双脉冲转移轨道目标函数 $f(t_0, \Delta t)$，枚举步长为 1 天。由图中可看出，目标函数呈现出多极值即多峰多谷的特性。

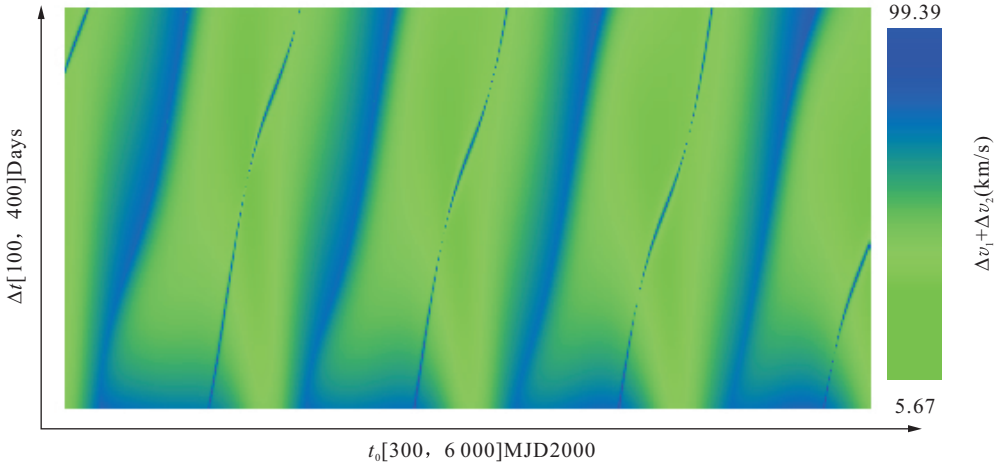


图 4-3 地球-火星双脉冲变轨问题

对于给定的发射时机范围 $t_0 \in [t_{0\min}, t_{0\max}]$（MJD2000，天）；$\Delta t$ 的范围 $\Delta t \in [\Delta t_{\min}, \Delta t_{\max}]$（天）；该优化问题的复杂性主要呈现以下几个方面特性：

(1)优化范围大，优化空间呈现二维性质，大小为：

$$R_{\text{two-thrust}} = [t_{0\min}, t_{0\max}] \times [\Delta t_{\min}, \Delta t_{\max}] \tag{4-12}$$

(2)精度要求高，求解精度的要求也直接影响优化性能，若精确到秒，则等价于优化范围扩大到：

$$R_{\text{two-thrust}} = [t_{0\min}, t_{0\max}] \times [\Delta t_{\min}, \Delta t_{\max}] \times 86\ 400^2 \tag{4-13}$$

(3)目标函数非线性，对决策向量的评价，需要求解 Lambert 问题，目标函数 $f(t_0, \Delta t)$ 呈现高度的非线性。

4.2.3　MPGA+DSM 和 MGA+DSM 优化问题复杂度

带机动的多借力飞行(MPGA)问题的优化,在目标函数求解上可以认为是多段 Lambert 问题的组合,而在优化空间的划分上,则可以认为是类似多段双脉冲交会问题的空间乘积。

考虑行星序列长度为 n 的 MPGA 任务,其优化问题的空间复杂度可描述如下:

$$R_{\mathrm{MPGA}} = 2^{n-1} \times [t_{0\min}, t_{0\max}] \times [\Delta t_{1\min}, \Delta t_{2\max}] \times \cdots \times [\Delta t_{n-1\min}, \Delta t_{n-1\max}] \tag{4-14}$$

其中 2^{n-1} 项为:每次求解一段 Lambert 轨道,都需要尝试长程和短程轨道。

不难看出,多借力飞行优化问题随着行星序列的增长,其空间复杂度呈指数级的生长。因此在优化过程中对优化空间的修剪是非常有必要的。

对于 MPGA+位置模型 DSM 的优化问题,设脉冲机动次数为 n,其中 m 次为 DSM 机动,那么 MPGA+DSM 优化问题空间复杂度为:

$$R_{\mathrm{MPGA+DSM}} = R_{\mathrm{MPGA}} \times \prod_{1}^{m} ([X_{i\min}, X_{i\max}] \times [Y_{i\min}, Y_{i\max}] \times [Z_{i\min}, Z_{i\max}]) \tag{4-15}$$

式中:$[X_{i\min}, X_{i\max}]$、$[Y_{i\min}, Y_{i\max}]$ 和 $[Z_{i\min}, Z_{i\max}]$ 表示第 i 次 DSM 的位置约束。

为便于计算,将 MGA+速度模型 DSM 的优化问题的部分参数归一化:

$$u(X) \in [-\pi/2, \pi/2] \Rightarrow u(X) \in [0,1]$$
$$v(X) \in [0, 2\pi] \Rightarrow v(X) \in [0,1]$$
$$\theta(X) \in [-\pi, \pi] \Rightarrow \theta(X) \in [0,1]$$

结合归一化后区间中的精度需求,设区间宽度为 R_u,则 MGA+DSM 优化问题的空间复杂度可估算如下:

$$R_{\mathrm{MGA+DSM}} = 2^{n-1} \times [t_{0\min}, t_{0\max}] \times \prod_{i=1}^{n-1} [\Delta T_{i\min}, \Delta T_{i\max}] \times \prod_{i=2}^{n-1} [r_{p_{i\min}}, r_{p_{i\max}}] \times R_u^{2(n-2)+3} \tag{4-16}$$

式中:$R_u^{2(n-2)+3}$ 为归一化后的参数 $u(X)$、$v(X)$、$\theta(X)$ 和 η 的优化空间;

2^{n-1} 为 Lambert 轨道段的数量(第一次 DSM 前的一段是速度模型确定的轨道,之后每两颗行星之间轨道分为一段 Kepler 轨道和一段 Lambert 轨道)。

第五章　行星际轨道优化的差分演化算法

§5.1　差分演化算法

本书没有对优化算法的选取做详细的分析，而只选取了遗传算法和差分演化算法来进行优化设计，原因在于 Myatt(2003)已经指出，对于航天任务优化设计技术，差分演化算法具有比较好的效果，此处使用遗传算法仅仅是为了与差分演化算法做比较。

下面先给出遗传算法和差分演化算法的基本原理与实现方法，随后给出实验结果。

5.1.1　遗传算法

遗传算法(Genetic Algorithm，GA)是由美国 Michigan 大学的 John Holland 教授于 1975 年构建的，它是模拟自然界生物进化机制的一种算法，即遵循“适者生存，优胜劣汰”的原则，也就是寻优过程中有用的保留，无用的则去除。在科学和生产实践中表现为，在所有可能的解决方法中找出最符合该问题所要求条件的解决方法，即找出一个最优解。遗传算法的思想来源于达尔文的进化论、孟德尔的群体遗传学说和魏茨曼的物种选择学说，其基本思想是模拟自然界遗传机制和生物进化论而形成的一种过程搜索最优解的算法。

遗传算法的特点是对参数进行编码运算，几乎不需要所求问题的任何信息，仅需要目标函数的信息。不受搜索空间是否连续或目标函数可微的限制就可找到最优解。依据它的并行性，非常适用于大规模并行计算机。因此，遗传算法广泛地应用于自动控制、计算科学、模式识别、工程设计、智能故障诊断、管理科学和社会科学领域，适用于解决复杂的非线性和多维空间寻优问题(陈国良等，1996；刘勇等，1998)。

与传统搜索算法不同，遗传算法从一组随机产生的初始解(称为群体)开始搜索过程，群体中的每个个体是问题的一个解(称为染色体)，这些染色体在后续迭代中不断进化(称为遗传)。

遗传算法主要通过交叉、变异、选择运算实现。交叉或变异运算生成下一代染色体(称为后代)，染色体的好坏用适应度来衡量，根据适应度的大小从上一代和后代中选择一定数量的个体，作为下一代群体，再继续进化，这样经过若干代之后，算法收敛于最好的染色体，它很可能就是问题的最优解或次优解。遗传算法中使用适应度概念来度量群体中的各个个体的好坏。

GA 的基本思想是对需要优化的问题的参数进行编码，然后由若干个编码后的参数形成一个初始种群作为优化问题的候选解，使用选择、交叉和变异三种算子进行操作，不断迭代优化，直到满足停机条件，如找到最优解和迭代到最大代数。

遗传算法的计算流程如图 5－1 所示。从图中可以看出，遗传算法是一种种群型操作，该操作以种群中的所有个体为对象。具体求解步骤如下（吉根林，2004；肖伟等，2004）。

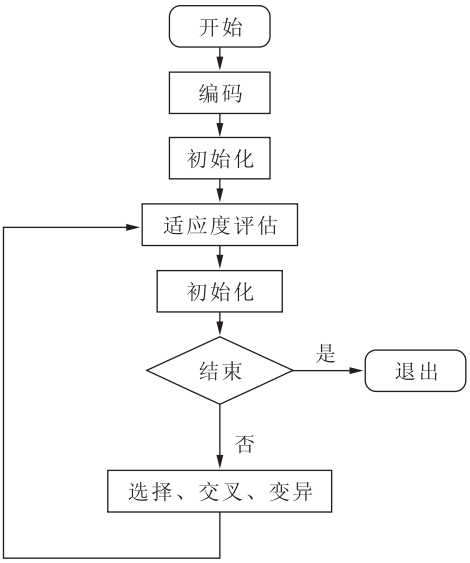


图 5－1　遗传算法的计算流程图

（1）编码：依据优化问题的特点确定使用何种码制进行编码，然后将问题参数编码形成基因链，即染色体。每一个染色体代表一个个体，表示优化问题的一个解。

（2）初始化：随机产生一个规模为 P 的初始种群，其中每个个体为一定长度（或不定长）的染色体，该群体代表优化问题的一些可能解的集合。

（3）适应度评估：计算种群中每个个体的适应度，适应度为群体进化时的选择提供了依据。一般来说，适应度越高，表示解越好。适应度函数一般可以根据目标函数而定。

（4）选择：根据每个个体的适应度，依据一定的选择标准，进行选择操作，产生新的个体加入到下一代群体中，一般个体被选择的概率与其适应度优劣成正比。

（5）交叉：从种群中随机选择两个染色体，按一定的交叉概率进行基因交换。交换位置的选取是随机的，也可以是多个基因位的交换。

（6）变异：从种群中随机地选择一个染色体，按一定的变异概率进行基因变异。

GA 的搜索能力主要是由选择与交叉操作体现的，变异算子则保证了算法能搜索到问题空间的每一点，从而使算法具有全局最优性，进一步增强了 GA 的搜索能力。

(7)若发现最优解或迭代到最大代数,则算法停止,否则转(3),对产生的新一代群体进行重新评估、选择、交叉、变异操作,如此循环往复,使群体中最优个体的适应度和平均适应度不断提高。

使用遗传算法求解问题,需要根据优化问题本身的特点,确定种群规模、交叉概率和变异概率等几个重要的参数,参数的选取直接影响到算法的性能。

5.1.2 差分演化算法

差分演化算法(Differential Evolution,DE)是 Rainer Storn 和 Kenneth Price 在 1995 年为求解有关切比雪夫多项式的问题提出来的,是一类基于群体差异的演化算法。最初,主要目的是用差分演化算法来求解连续全局优化问题。其基本思想是应用当前种群中个体的差来重组得到中间种群,然后通过父子之间的锦标赛制的竞争获得新一代种群。

差分演化算法最新颖的特征是它的变异操作,当选定一个个体后,算法通过在该个体上加上两个个体带权的差来完成变异。在算法迭代的初期,因为种群中个体的差较大,从而这样的变异操作会使算法本身具有较强的全局搜索能力;而到迭代的后期,当趋于收敛的时候,种群中个体的差较小了,这也使得算法具有较强的局部搜索能力。这种新颖的变异操作也使得该算法在求解函数优化等问题上有着其他同类方法不可比拟的优点。主要优点可以总结为以下三点(胡中波,2006):①待定参数少;②不易陷入局部最优;③收敛速度快。

差分演化是一种基于实数编码的演化算法,算法的基本思想及整体构架与遗传算法相类似,从一代种群到下一代种群都要经过变异、交叉、选择等操作,也一样有几个至关重要的参数必须事先确定。与遗传算法的主要区别在于变异操作上,差分演化的变异操作是基于染色体的差异向量进行的,其余操作和遗传算法类似。下面通过求解非线性函数 $f(x_1,x_2,\cdots,x_n)$ 的最小值问题,x_i 满足 $x_i^L \leqslant x_i \leqslant x_i^U$,$i=1,2,\cdots,n$,来介绍差分演化算法的操作过程(Srinivas,*et al*,1994):

令 $x_j(t)$ 是第 t 代的第 j 个染色体,则:

$$x_j(t)=(x_{j1}(t),x_{j2}(t),\cdots,x_{jn}),\quad j=1,2,\cdots,M;t=1,2,\cdots,t_{\max} \tag{5-1}$$

式中:n 为染色体的长度,即变量的个数;

M 为群体规模;

$t_{\max}$ 为最大的进化代数。

第一步:生成初始种群

在 n 维空间里随机产生满足约束条件的 M 个染色体,实施措施如下:

$$x_{ji}(0)=rnd_{ji}(0,1)(x_i^U-x_i^L)+x_i^L,\quad j=1,2,\cdots,M \tag{5-2}$$

式中:x_i^U,x_i^L 分别为第 i 个变量的上界和下界,$i=1,2,\cdots,n$;

$rnd_{ij}(0,1)$ 为[0,1]之间的随机小数。

第二步：变异操作

从群体中随机选择 3 个染色体 x_{p1}, x_{p2}, x_{p3} 且 $j \neq p1 \neq p2 \neq p3$，则：

$$h_{ji}(t+1) = x_{p1i}(t) + F \times [x_{p2i}(t) - x_{p3i}(t)] \tag{5-3}$$

式中：$x_{p2i}(t) - x_{p3i}(t)$ 为差异化向量；

F 为缩放因子。

第三步：交叉操作

交叉操作是为了增加种群的多样性，具体操作如下：

$$v_{ji}(t+1) = \begin{cases} h_{ji}(t+1), & rand_{ji} \leqslant CR \text{ and } i < n \\ x_{ji}, & \text{otherwise} \end{cases} \tag{5-4}$$

式中：$rand_{ji}$ 为[0,1]之间的随机小数；

CR 为交叉概率，$CR \in [0,1]$。

第四步：选择操作

为了决定 $x_j(t)$ 是否成为下一代的成员，向量 $v_j(t+1)$ 和目标向量 $x_j(t)$ 对评价函数进行比较：

$$x_j(t+1) = \begin{cases} v_j(t+1), & f(v_{j1}(t+1), \cdots, v_{jn}(t+1)) < f(x_{j1}(t), \cdots, x_{jn}(t)) \\ x_j(t), & f(v_{j1}(t+1), \cdots, v_{jn}(t+1)) \geqslant f(x_{j1}(t), \cdots, x_{jn}(t)) \end{cases} \tag{5-5}$$

反复执行第二步到第四步的操作，直至达到最大的进化代数 $t_{\max}$。

§5.2　实验结果

5.2.1　双脉冲＋单次位置模型 DSM

双脉冲＋单次位置模型的优化问题决策向量：

$$X = \{t_0, \Delta t, \alpha, \vec{P}_{\mathrm{DSM}}\} \tag{5-6}$$

式中：$\alpha \in [0,1]$ 为航天器在 DSM 发生前的一段 Lambert 轨道时长占总 Δt 中的比例；决策向量 X 是式(4-2)的变形，转换到式(4-2)的形式为：

$$X = \{t_0, \mathrm{eph}(2, t_0), \alpha\Delta t, \vec{P}_{\mathrm{dsm}}, (1-\alpha)\Delta t, \mathrm{eph}(3, t_0 + \Delta t)\} \tag{5-7}$$

式中：$\mathrm{eph}(\mathrm{Pl}, t_0)$ 为星历计算，详见 9.2 节，本实验采用 ESA 提供的星历模型(ACT，2008)，

$\mathrm{eph}(2, t)$ 即为计算地球在 t 时刻的位置；

$\vec{P}_{DSM}$ 为 DSM 位置。

发射时刻范围(MJD2000)：$t_0 \in [800, 3\,800]$。

表 5-1 中第一行为 ESA(Myatt，*et al*，2003)的最优解，方案 A 和方案 B 在总 Δv 上略占优势，但其转移时长过大，虽然没有什么实用价值，但却能说明非共面双脉冲转移轨道在过程中增加 DSM 的必要性。

表 5-1 地球-火星双脉冲+单次 DSM 优化结果

方案	发射时刻 t_0 (MJD2000)	转移时长 (天)	DSM 时刻 (比例)	DSM 位置 ($O_s x_s y_s z_s$)	总 Δv (km/s)	总 Δt (天)
ESA	1 253.7	202.9	NA	NA	5.667	[100～400]
DE-A	1 243.682	579.326	0.350	162 230 102.134, 140 397 524.849, −1 050 664.561	5.632	[100～600]
DE-B	1 247.264	400.000	0.491	161 631 817.084, 140 190 911.230, −1 163 778.203	5.635	[100−400]

5.2.2 双脉冲+单次速度模型 DSM

速度模型 DSM 的优化结果也表明,双脉冲+单次 DSM 的行星际转移轨道比直接双脉冲转移轨道更优。实验中使用 DE 算法进行优化,种群规模:200;运行代数:200;DE 策略:Best1Bin;缩放因子:0.9;变异概率:0.9。

双脉冲+单次速度模型 DSM 的决策向量:

$$X = \{t_0, v_\infty, u, v, \eta, \Delta t\} \tag{5-8}$$

参数说明见式(4-8)。在优化实验中,$\eta \in [0.1, 0.9]$,$u \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$,$v \in [-\pi, \pi]$。

地球-水星优化结果比较(表 5-2):

参数范围:$t_0 \in [800, 3\,800]$ MJD2000,$\Delta t \in [100, 600]$ Days,$v_\infty \in [0, 18]$(km/s)。

总 Δv 减少量:1.114%。

表 5-2 地球-水星双脉冲+DSM 优化结果

参数 / 最优类别	t_0	Δt	η	v_∞	u	v	Δv
双脉冲	1 957.269	104.884	NA	NA	NA	NA	17.558
双脉冲+DSM	1 957.355	455.723	0.785 2	0.210	4.481	5.623	17.350

地球-金星优化结果比较(表 5-3):

参数范围:$t_0 \in [800, 3\,800]$ MJD2000,$\Delta t \in [100, 600]$ Days,$v_\infty \in [0, 6]$(km/s)。

总 Δv 减少量:9.220%。

表 5-3　地球-金星双脉冲+DSM 优化结果

参数 / 最优类别	t_0	Δt	η	v_∞	u	v	Δv
双脉冲	2 715.316	133.275	NA	NA	NA	NA	6.117
双脉冲+DSM	3 264.054	265.342	0.558	2.356	1.198	6.062	5.553

地球-火星优化结果比较(表 5-4)：

参数范围：$t_0 \in [800,3\ 800]$ MJD2000，$\Delta t \in [100,600]$ Days，$v_\infty \in [0,6]$ (km/s)。

总 Δv 减少量：0.635%。

表 5-4　地球-火星双脉冲+DSM 优化结果 1

参数 / 最优类别	t_0	Δt	η	v_∞	u	v	Δv
双脉冲	1 253.436	203.666	NA	NA	NA	NA	5.669
双脉冲+DSM	1 244.364	537.659	0.375	2.777	4.583	2.761	5.633

地球-火星优化结果比较 2(表 5-5)：

参数范围：$t_0 \in [3\ 800,7\ 000]$ MJD2000，$\Delta t \in [100,600]$ Days，$v_\infty \in [0,6]$ (km/s)。

总 Δv 减少量：2.291%。

表 5-5　地球-火星双脉冲+DSM 优化结果 2

参数 / 最优类别	t_0	Δt	η	v_∞	u	v	Δv
双脉冲	6 705.511	204.468	NA	NA	NA	NA	5.760
双脉冲+DSM	6 355.994	581.666	0.646	0.288	4.580	2.729	5.628

地球-土星优化结果比较(表 5-6)：

参数范围：$t_0 \in [800,3\ 800]$ MJD2000，$\Delta t \in [1000,6\ 000]$ Days，$v_\infty \in [0,20]$ (km/s)。

总 Δv 减少量：8.488%。

表 5-6　地球-土星双脉冲+DSM 优化结果

参数 / 最优类别	t_0	Δt	η	v_∞	u	v	Δv
双脉冲	3 636.217	2 201.340	NA	NA	NA	NA	16.954
双脉冲+DSM	3 718.195	3 656.646	0.387	0.036	2.281	4.191	15.515

5.2.3 EVVEJS

以 EVVEJS 为例，其中决策向量空间，即式(4－2)中的 X 取值范围为：

$$t_0 \in [-1\ 000, 0],\ \Delta t_1 \in [30, 400],\ \Delta t_2 \in [100, 470]$$
$$\Delta t_3 \in [30, 400],\ \Delta t_4 \in [400, 2\ 000],\ \Delta t_5 \in [1\ 000, 6\ 000] \tag{5-9}$$

并且满足约束条件：

$$r_{p1} > 6\ 351.8\text{km}, r_{p2} > 6\ 351.8\text{km},$$
$$r_{p3} > 6\ 778.1\text{km}, r_{p4} > 671\ 492\text{km} \tag{5-10}$$

对于该问题，根据 Becerra 等(2005)中的分析可知，当前已知的最好结果为4.870km/s，并且该问题的目标函数，即式(4－5)的全局最小值具有一个较小的吸引盆，最小值是伴生的，从而使得搜索很难达到全局最优，Becerra 等(2005)中使用改进的差分演化算法，选取种群规模为 40，演化代数 2 000 代，执行了 20 次实验，结果 19 次找到了次优解 5.225 7km/s。Victor 等(2006)中给出了该结果的轨道示意图，即图 5－2，而仅有一次得到的结果比较接近 4.870km/s。另外，欧空局官方网站上给出了使用改进的差分演化算法，求得的最好结果为 4.930 722 969 423。表 5－7 给出了本书结果和文献中的结果。

表 5－7 本书结果与文献结果比较

		$t_0, \Delta t_1, \Delta t_2, \Delta t_3, \Delta t_4, \Delta t_5$	$v_0, v_1, v_2, v_3, v_4, v_5$	$r_{p1}, r_{p2}, r_{p3}, r_{p4}$	$\min(f)$
ESA (图 5－4)		−789.811 721 798 620 158.302 027 105 278 449.385 873 819 743 54.748 968 433 966 1 024.362 058 469 180 4 552.307 968 055 420	2.754 635 857 391 1.090 646 468 834 0.615 767 825 576 0.000 000 000 000 0.000 000 000 000 0.469 672 817 621	351.799 996 179 795 8 881.505 617 379 225 6 778.100 000 356 682 833 991.160 791 673 810	4.930 722 969 423
本书	GA	−806.000 000 000 000 200.200 000 000 000 421.900 000 000 000 56.270 000 000 000 1 014.400 000 000 000 4 465.000 000 000 000	4.728 075 840 606 0.153 904 148 794 0.377 747 492 557 0.133 587 141 350 0.038 052 791 329 0.473 358 271 810	12 320.761 271 124 080 12 657.552 846 682 298 6 810.686 394 663 274 872 375.131 548 693 520	5.904 725 686 446
	DE	−789.765 969 545 302 158.315 220 636 331 449.385 882 289 005 54.709 938 125 694 1 024.760 002 461 615 4 552.909 411 933 883	2.754 598 861 690 1.092 416 952 089 0.613 944 512 224 0.000 000 001 814 0.000 000 001 171 0.469 747 973 358	6 351.800 062 405 529 8 866.321 828 342 687 6 778.100 023 156 845 832 794.986 804 036 310	4.930 708 302 346

从表 5－7 可以看出，DE 明显优于 GA。我们改进了基本的 DE 算法，在算法搜索的后期，根据对问题的已知信息，采取启发式的搜索方式，得到了满意的效果。参照文献（Becerra，*et al*，2005），进行了 20 次实验，种群规模 200，演化代数 2 000，得到最优解 4.930 708 302 346，而得到的次优解为 5.303 422 371 912。图 5－3 至图 5－5 给出了相应结果的仿真示意图。

虽然本书结果优于文献中给出的结果，但是仔细分析发现，本书结果中 v_3、v_4 均不为零，即需要对航天器深空机动；而文献中的结果均为零，不需要深空机动。从航天器工程的角度考虑，点火次数会影响航天器寿命，点火次数越少越好。

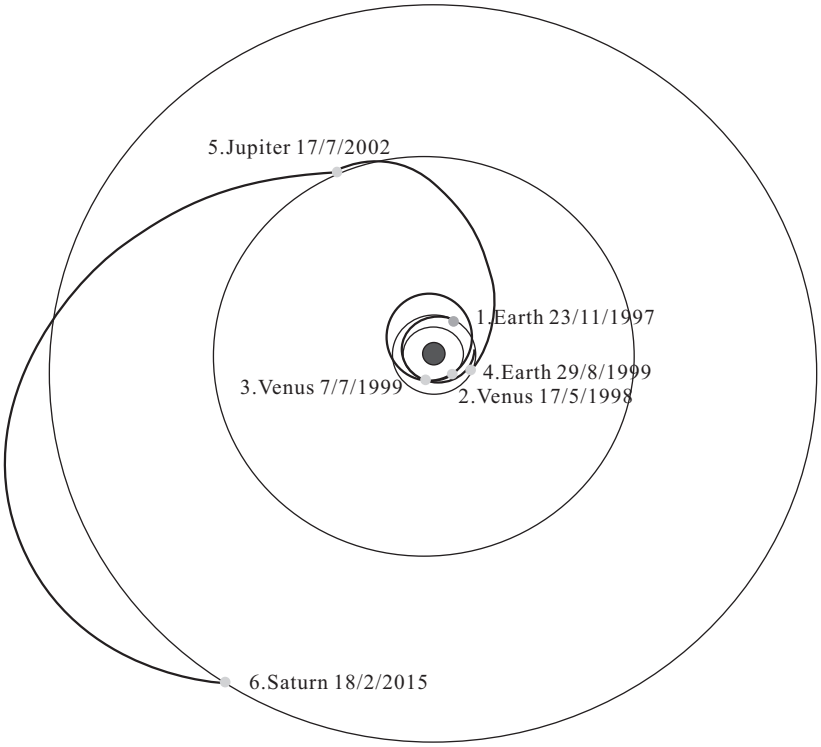


图 5－2　min(f)＝5.225 7km/s

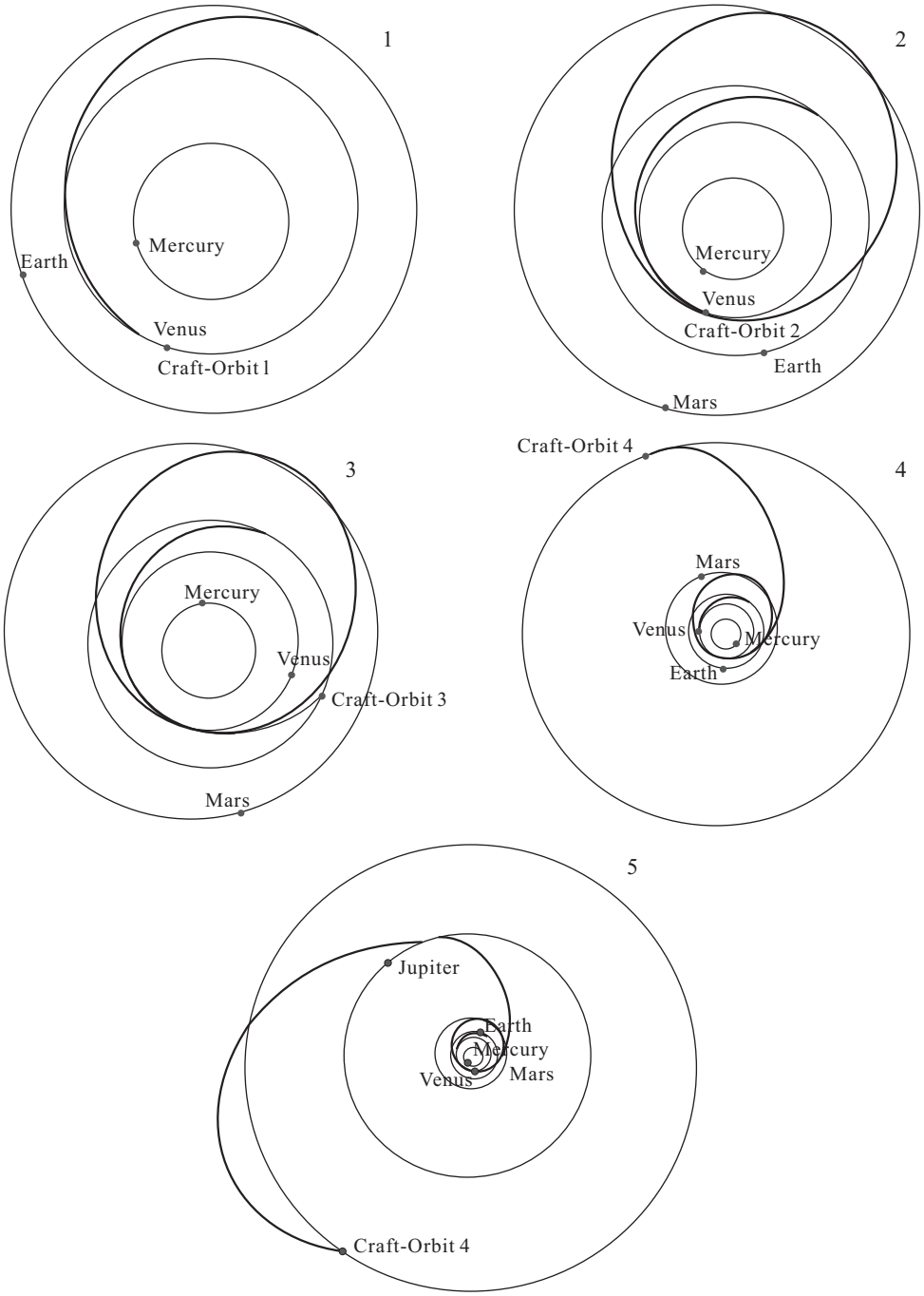


图 5－3　min(f)＝4.930 708 302 346km/s

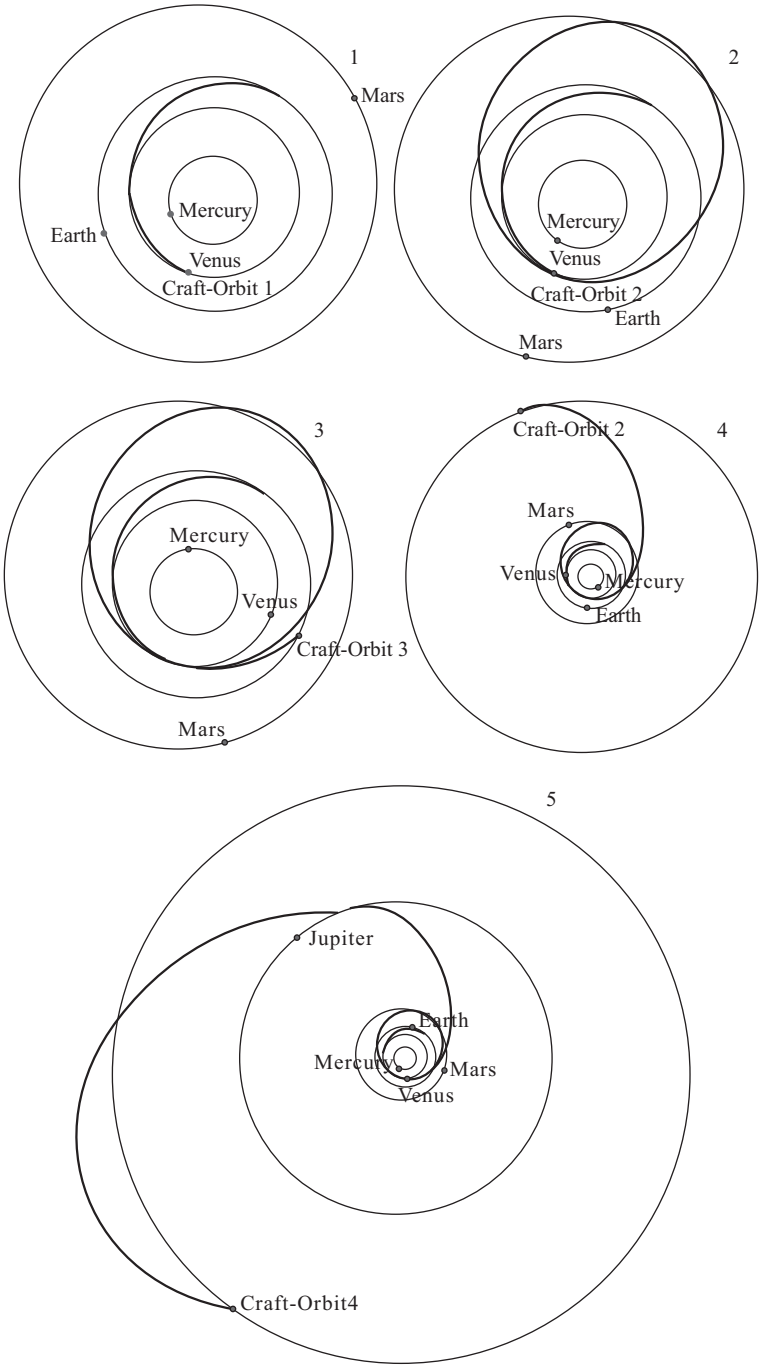


图 5－4　min(f)＝4.930 722 969 423km/s

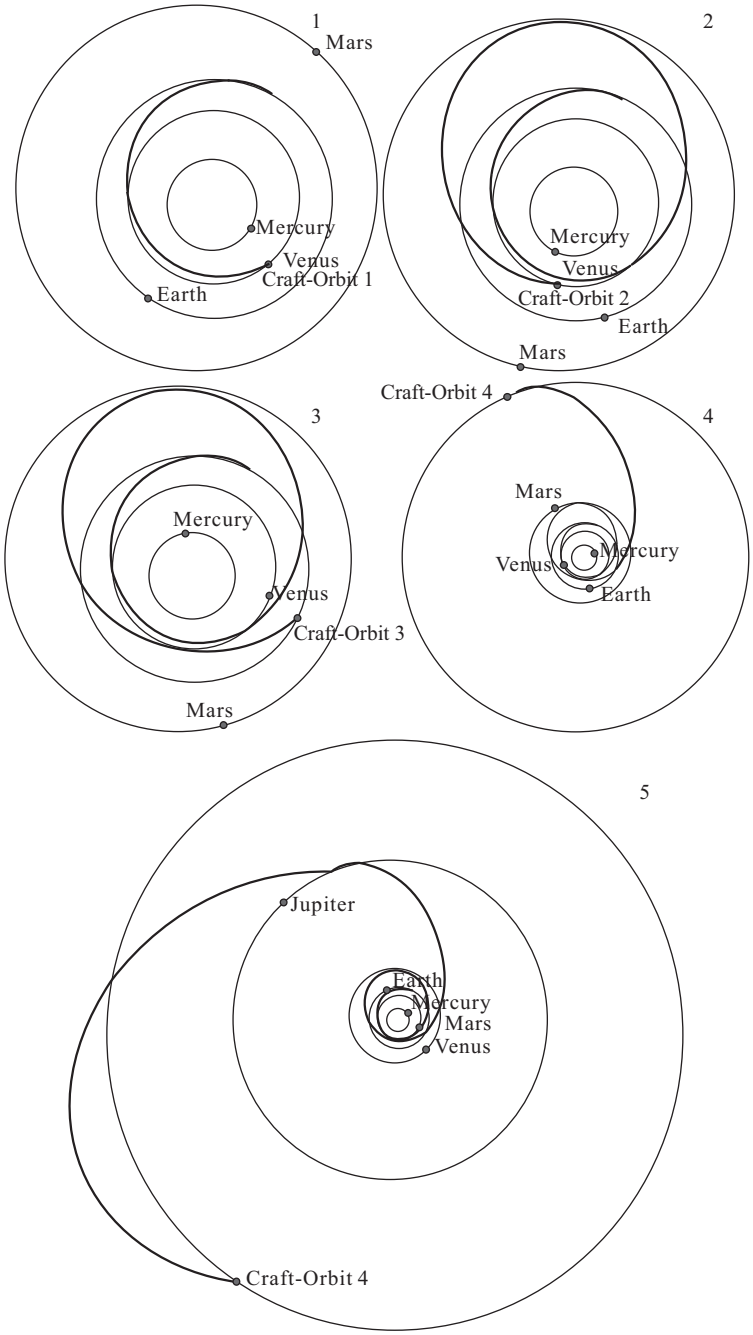


图 5－5　min(f)＝5.303 422 371 912km/s

第六章　行星际轨道优化设计的 SQP 算法

§6.1　SQP 方法

SQP 方法，即序列二次规划，是解决非线性约束问题的最有效的方法。此方法是在牛顿线性搜索上发展起来的，牛顿法是用于解决无约束非线性问题的好方法，它可以看作是寻求作为最优性必要条件的目标函数梯度零点的一种方法。SQP 方法是由此得到启发，将牛顿法用于求解约束问题最优性必要条件的 $K-T$ 方程组导出来的一种算法。

这里着重对 SQP 方法进行研究，拟牛顿法在文献（王于，2006）中已经详细介绍，不再赘述。

由于地球-火星转移轨道优化设计问题的目标函数不是一个简单的非线性函数，它还有一些关于能量和时间的约束条件。对于解决这种非线性的带约束条件的优化问题，SQP 可以很有效地解决（徐成贤等，2002；袁亚湘等，1997；席少霜，1992；Fukushima，1986；McCormick，1983；Goldberg，1989）。顾名思义，序列二次规划法主要是用二次规划（QP）来顺序求解搜索方向而得名。在国外，SQP 是一种常用的解决卫星交会的最优化方法。由于这是一个纯数学的面向过程的算法，涉及的数学知识很多，很复杂，这里仅选择与推导相关的定理来论述。

6.1.1　关于 SQP 的定理和公式

首先介绍若干关于非线性带约束规划的定义（徐成贤等，2002）。

定义 6.1　非线性带约束规划

$$\begin{aligned} &\min \quad f(x) \qquad x \in R^n \\ &\text{s.t.} \quad c_i(x) \leqslant 0, \quad i \in \{1,2,\cdots,m\} \end{aligned} \tag{6-1}$$

其中：$c_i: R^n \to R$，$(i = 1,2,\cdots,m)$是光滑函数。

记可行域为：

$$D = \{x \in R^n \mid c_i(x) \leqslant 0,\ i \in \{1,2,\cdots,m\}\}。$$

先考虑等式约束问题：

$$\begin{aligned} &\min \quad f(x) \quad x \in R^n \\ &\text{s.t.} \quad c_i(x) = 0, \quad i \in E = \{1,2,\cdots,l\} \end{aligned} \tag{6-2}$$

定理 6.1 （一阶必要条件）

若 1）等式约束问题(6－2)局部最优解是 x^*；

2）$f(x)$ 与 $c_i(x)$,$(i=1,2,\cdots,l)$ 在 x^* 的某个邻域内连续可微；

3）$\nabla c_i(x^*)$,$(i=1,2,\cdots,l)$ 线性无关；

则存在一组不全为零的实数 $\lambda_1^*,\cdots,\lambda_l^*$ 使得

$$\nabla f(x^*) - \sum_{i=1}^{l} \lambda_i^* \nabla c_i(x^*) = 0 \tag{6-3}$$

等式约束问题(6－2)的最优解所对应的一阶必要条件就是非线性方程组(6－3),但非充分条件。

定理 6.2 （二阶充分条件)在等式约束问题(6－2)中,若

1）$f(x)$ 与 $c_i(x)$,$(i=1,2,\cdots,l)$ 是二阶连续可微函数

2）存在于使 Lagrange 函数的梯度为零,即：

$$\nabla L(x^*,\lambda^*) = 0$$

3)对任意非零向量 $s \in R^n$ 且

$$s^T \nabla c_i(x^*) = 0, i = 1,2,\cdots,l \text{ 均有 } s^T \nabla_x^2 L(x^*,\lambda^*)s > 0$$

则 x^* 是问题(6－2)的严格局部极小点。

定义 6.2 有效集的概念

若不等式约束问题(6－1)的一个可行点 x^* 使某个不等式约束 $c_i(x) \leqslant 0$,变成等式 $c_i(x)=0$,则该不等式约束称为关于 x^* 的有效约束。否则,若对某个 k 使得 $c_k(x) < 0$,则该不等式约束称为关于 x^* 的非有效约束。称所有在 x^* 处的有效约束的指标组成的集合 $I = I(x^*) = \{i \mid c_i(x^*) = 0\}$ 为 x^* 处的有效指标集,简称有效集。

定理 6.3 （Kuhn－Tucker 一阶必要条件）

在不等式约束问题(6－1)中,若：

1）x^* 为局部最优解,有效集 $I^* = \{i \mid c_i(x^*) = 0, i = 1,2,\cdots,m\}$；

2）$f(x)$ 与 $c_i(x)$,$(i=1,2,\cdots,m)$ 在 x^* 的某邻域内连续可微；

3）对于 $i \in I^*$ 的 $\nabla c_i(x^*)$ 线性无关(此条件叫 Kuhn－Tucker 约束规范),则存在向量 $\lambda^* = (\lambda_1^*,\cdots,\lambda_m^*)^T$ 使

$$\begin{aligned} &\nabla f(x^*) - \sum_{i=1}^{m} \lambda_i^* \nabla c_i(x^*) = 0 \\ &\lambda_i^* \nabla c_i(x^*) = 0, \ i = 1,2,\cdots,m \\ &\lambda_i^* \geqslant 0, \ i = 1,2,\cdots,m \end{aligned}$$

上面已经说明了必要条件,最后说明凸规划问题最优解的充分条件。设凸规划问题：

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & c_i(x) = a_i^T x + b_i = 0, i \in E = \{1,2,\cdots,l\} \\ & c_i(x) \leqslant 0, i \in I = \{l+1,l+2,\cdots,m\} \end{aligned} \tag{6-4}$$

其中 $f(x)$ 和 $c_i(x)$ 均为凸函数。

定理 6.4　（凸规划问题的充分条件）

设凸规划问题(6－4)中 $f(x)$ 和 $c_i(x)(i \in I)$ 为可微函数，若 x^* 为问题(6－4)的 $K-T$ 点，则 x^* 是问题(6－4)的整体最优解。

前面给出了有关最优化问题的概念以及定理、定义，相关证明参考相关文献。

6.1.2　Jacobi 迭代法

在数学规划的相关算法中，求解线性方程组是一个必不可少的环节，这里用雅可比Jacobi迭代法来求解线性方程组。

1. Jacobi 迭代法的数学原理

Jacobi 迭代法的基本思想就是迭代，具体做法是从方程组 $A\vec{x} = \vec{b}$ 推导出一个迭代公式，从可行域随意选取一个初始向量 $\vec{x}^{(0)}$ 代入推导出的迭代公式，解出 $\vec{x}^{(1)}$，再将 $\vec{x}^{(1)}$ 代入同一迭代公式，得到 $\vec{x}^{(2)}$，以此类推，可以求得一个向量的序列 $\{\vec{x}^{(k)}\}$。若 $\{\vec{x}^{(k)}\}$ 收敛，其极限就是原方程组的解。

详细说明如下，假设有方程组 $\sum_{j=1}^{n} a_{ij}x_j = b_i, i = 1,2,\cdots,n$，可以简单改写为：

$$Ax = b \tag{6-5}$$

上式中，A 为非奇异矩阵，$a_{ii} \neq 0(i = 1,2,\cdots,n)$。将矩阵 A 分为 D、L 和 U 三个子矩阵，分别表示如下：

$$D = \begin{Bmatrix} a_{11} & & & \\ & a_{22} & & \\ & & \ddots & \\ & & & a_{nn} \end{Bmatrix}$$

$$L = -\begin{Bmatrix} 0 & & & \\ a_{21} & 0 & & \\ a_{31} & a_{32} & 0 & \\ \cdots & & & \\ a_{n1} & a_{n2} & \cdots & 0 \end{Bmatrix}$$

$$U=-\begin{Bmatrix} 0 & a_{12} & a_{13} & \cdots & a_{1n} \\ & 0 & a_{23} & \cdots & a_{3n} \\ & & & & \cdots \\ & & & & 0 \end{Bmatrix}$$

将式(6-5)的第 i 个方程除以 a_{ii},然后移项可以得到等价的方程组:

$$x_i=\frac{1}{a_{ii}}(b_i-\sum_{\substack{j=1\\ j\neq i}}^{n}a_{ij}x_j),\quad i=1,2,\cdots,n \tag{6-6}$$

记为: $x=Bx+f$

其中: $B=I-D^{-1}A=D^{-1}(L+U)$, $f=D^{-1}b$

对方程组(6-6)使用迭代方法:

给定一个初始向量 $\vec{x}^{(0)}$,利用下式迭代:

$$x_i^{(k+1)}=\frac{1}{a_{ii}}(b_i-\sum_{\substack{j=1\\ j\neq i}}^{n}a_{ij}x_j^{(k)}) \tag{6-7}$$

迭代的矩阵形式为:

$$X^{(k+1)}=BX^{(k)}+f \tag{6-8}$$

其中,B 是迭代矩阵,用 Gauss 列主消元法求解其迭代的初始向量。

2. Jacobi 迭代法的算法流程

下面具体介绍雅可比(Jacobi)迭代法解方程组算法流程。设方程组 $A\vec{x}=\vec{b}$ 系数矩阵的主对角线元素均不为零,即 $a_{ii}\neq 0(i=1,2,\cdots,n)$ 。迭代方式如下:

S1:选取初始向量 $\vec{x}=(x_1^{(0)},x_2^{(0)},\cdots,x_n^{(0)})^T$,令 $k=0$;

S2:计算下式:

$$x_i^{(k+1)}=\frac{1}{a_{ii}}(b_i-\sum_{\substack{j=1\\ j\neq i}}^{n}a_{ij}x_j^{(k)})$$

其中,$i=1,2,\cdots,n$;

S3:如果有 $\sum_{i=1}^{n}|x_i^{(k+1)}-x_i^{(k)}|<\varepsilon$,则求得方程组的解 $\vec{x}^{(k+1)}$,算法终止;否则执行 S4;

S4:若 $k\geqslant M$,则表示该方程组无法收敛,程序终止;否则令 $k=k+1$,转 S2。

其中,M 为最大迭代次数,ε 为容许误差。

6.1.3 二次规划(QP)及其有效集法

二次规划(QP)是序列二次规划(SQP)的核心,在 SQP 方法中每一轮迭代都需要求解子二次规划问题。本节针对这个问题进行介绍。

1. 二次规划的算法原理

对于目标函数为二次函数、约束条件为线性函数的数学规划问题统称为二次规划问题

(Quadratic Programming)，一般的求解形式如下：

$$\begin{aligned} \min \quad & f(x) = \frac{1}{2}x^T G x + C^T x \\ \text{s.t.} \quad & c_i(x) = a_i^T x + b_i = 0, \quad i \in E = \{1,2,\cdots,l\} \\ & c_i(x) = a_i^T x + b_i \leqslant 0, \quad i \in I = \{l+1,l+2,\cdots,m\} \end{aligned} \tag{6-9}$$

式中：G 称为 Hesse 矩阵，它是 n 阶对称矩阵；

C 为 n 维向量。

G 的性质决定了二次规划解的性质，若 G 正定，则称式(6-9) 为严格凸二次规划，其产生的解是效益函数的下降方向，进而问题有唯一解；若 G 正半定，则称式(6-9) 为凸二次规划，此时问题能否求解不定；若 G 不定，则称式(6-9)为非凸二次规划，那么可能有多个局部解，从而造成求解困难。由于 SQP 方法中需要在每一步迭代中保证 Hesse 矩阵的正定性，从而需要对其进行修正，因此这里仅讨论严格凸二次规划问题。对于严格凸二次规划问题来讲，Kuhn-Tucker 条件就是最优解的充要条件。

定理 6.5　若 x^* 满足 $K-T$ 条件，即存在乘子向量 $\lambda^* = (\lambda_1^*, \lambda_2^*, \cdots, \lambda_m^*)^T$，使得

$$\begin{aligned} & Gx^* + C - \sum_{i \in E} \lambda_i^* a_i - \sum_{i \in I} \lambda_i^* a_i = 0 \\ & c_i(x) = a_i^T x + b_i = 0,\ i \in E = \{1,2,\cdots,l\} \\ & c_i(x) = a_i^T x + b_i \leqslant 0,\ i \in I = \{l+1,l+2,\cdots,m\} \\ & \lambda_i^* \geqslant 0,\ i \in I \\ & \lambda_i^* = 0,\ i \in I/I^* \end{aligned} \tag{6-10}$$

其中 I^* 为 x^* 处的有效集，则称点 x^* 是严格凸二次规划(QP)的严格全局最优解，这是一个充要条件。

下面分两种情况讨论如何求解严格凸二次规划问题。

(1)仅含等式约束的严格凸二次规划问题

假设此类二次规划问题形如：

$$\begin{aligned} \min \quad & f(x) = \frac{1}{2}x^T G x + C^T x \\ \text{s.t.} \quad & c_i(x) = a_i^T x + b_i = 0,\ i \in E = \{1,2,\cdots,l\} \end{aligned} \tag{6-11}$$

其中，$A = (a_1 \cdots a_l)$ 的秩为 l，则由定理 6.5 可知，x 为严格凸二次规划问题(6-11)的全局最优解的充分必要条件是：

$$\begin{aligned} & Gx + C - \sum_{i \in E} \lambda_i a_i = 0 \\ & c_i(x) = a_i T x + b_i = 0,\ i \in E \end{aligned}$$

做替换，令 $b = (b_1, \cdots, b_l)^T$，$\lambda = (\lambda_1, \cdots, \lambda_l)^T$，则充要条件可以写成向量矩阵方程组形式：

$$\begin{Bmatrix} G & -A \\ G^T & 0 \end{Bmatrix} \begin{Bmatrix} x \\ \lambda \end{Bmatrix} = \begin{Bmatrix} -C \\ b \end{Bmatrix} \tag{6-12}$$

由于是严格凸二次规划，所以 G 正定，A 的秩为 l，因此 A 列满秩，由线性代数知识可以推得方程组(6－12)有唯一解$\begin{Bmatrix} x^* \\ \lambda^* \end{Bmatrix}$，其中 x^* 为问题(6－11)的最优解，λ^* 为相应的拉格朗日向量。

式(6－12)实际上就是求解线性方程组，用 Jacobi 迭代法就可以求解。

(2)一般严格凸二次规划的有效集法

定理 6.6 (有效集法的基础)设 x^* 是问题(6－9)的解，且在 x^* 处的有效集为 I^*，则 x^* 也是下列约束问题的最优解。

$$\begin{aligned} \min \quad & f(x) = \frac{1}{2}x^T G x + C^T x \\ \text{s.t.} \quad & c_i(x) = a_i^T x + b_i = 0, \qquad i \in I^* \end{aligned} \tag{6-13}$$

从定理 6.6 可知，若能找到严格凸二次规划问题(6－9)的某一个可行解 x_k，并且 x_k 恰好是问题(6－13)的最优解，则只要其拉格朗日乘子向量非负，由定理 6.5 可知 x_k 即为问题(6－9)的最优解。这样，问题可以由不等式约束问题简化等式约束问题，由方法(1)中求解方程组(6－12)，就可以求得最优解。这个思想就是有效集法的基本思想，下面详细说明。

设 x_k 为问题(6－9)的一个可行解，相应的有效集为 I_k，A_k 为 $A = (a_1, a_2, \cdots, a_m)$ 中对应 I_k 的子矩阵，求解二次规划问题：

$$\begin{aligned} \min \quad & f(x) = \frac{1}{2}x^T G x + C^T x \\ \text{s.t.} \quad & A_k^T x = b_k \end{aligned} \tag{6-14}$$

可以通过定理证明，问题(6－14)与下面二次规划问题(6－15)等价：

$$\begin{aligned} \min \quad & q(x) = \frac{1}{2}d^T G d + g_k^T d \\ \text{s.t.} \quad & A_k^T d = 0 \end{aligned} \tag{6-15}$$

其中：$x = x_k + d$，$g_k = \nabla f(x_k) = Gx_k + C$。

当二次规划问题(6－15)的最优解 $d_k = 0$ 时，则 x_k 为问题(6－14)最优解。若此时 x_k 对应的拉格朗日乘子向量非负，则由定理 6.5 知 x_k 为问题(6－9)的最优解从而迭代停止。当问题(6－15)的最优解 $d_k \neq 0$ 时，若 $x_k + d_k$ 为问题(6－9)的可行解，则令 $x_{k+1} = x_k + d_k$；否则以 d_k 为方向作线性搜索，以求得最好的可行点。由于 $f(x)$ 为凸二次函数，故此点必在边界上达到。记其步长为 $a_k > 0$，$x_{k+1} = x_k + a_k d_k$，则由可行性要求应该有：

$$a_i^T(x_k + a_k d_k) \leqslant b_i, \qquad i \notin I_k$$

移项得：

$$a_i^T x_k - b_i + a_k a_i^T d_k \leqslant 0, \qquad i \notin I_k$$

上式右端非正，当 $a_i^T d_k \geqslant 0$ 时，上式恒成立；反之若要求上式成立，则必须满足当 a_k 增加，只有当 $a_i^T d_k > 0$ 时，才有可能破坏约束条件，因此：

$$a_k \leqslant \frac{b_i - a_i^T x_k}{a_i^T d_k}, \qquad i \notin I_k \text{ 且 } a_i^T d_k > 0$$

故应该取：

$$a_k = \bar{a}_k = \min_{i \notin I_k} \left\{ \frac{b_i - a_i^T x_k}{a_i^T d_k} \mid a_i^T d_k > 0 \right\} = \frac{b_t - a_t^T x_k}{a_t^T d_k} \tag{6-16}$$

综合上述情况，取：

$$a_k = \min\{1, \bar{a}_k\} \tag{6-17}$$

其中：$\bar{a}_k$ 由式(6-16)确定，$x_{k+1} = x_k + a_k d_k$ 。

当 $a_k < 1$，即 $a_k = \bar{a}_k = \dfrac{b_t - a_t^T x_k}{a_t^T d_k}$ 时，则有某个 $t \notin I_k$ ，使得：

$$a_k = \bar{a}_k = \frac{b_t - a_t^T x_k}{a_t^T d_k}$$

进而得到：

$$a_t^T x_{k+1} = a_t^T x_k + a_k a_t^T d_k = b_t$$

因此，在 x_{k+1} 处增加一个有效约束，有效集变为 $I_k = I_k \cup \{t\}$ ；

当 $a_k = 1$ 时，则保持有效集不变进行下一轮的迭代。

若式(6-15)的最优解 $d_k = 0$，并且其拉格朗日乘子向量有负的分量，例如$(\lambda_k)_s < 0$，从 $K-T$ 条件可知 x_k 不是式(6-9)的最优解。此时应该找出一个下降可行方向 p_k ，满足：

$$g_k^T p_k < 0 \quad \text{且} \quad A_k^T p_k \geqslant 0$$

为了满足上述条件，选择的 p_k 必然使 $x_k + p_k$ 仍然位于除 s 个有效约束外的其他约束上，于是有：

$$a_j^T (x_k + p_k) = b_j, \quad j \in I_k, j \neq s$$
$$a_s^T (x_k + p_k) < b_s$$

变形为：

$$a_j^T p_k = 0, \quad j \in I_k, j \neq s \tag{6-18}$$

$$a_s^T p_k > 0 \tag{6-19}$$

由于 x_k 是式(6-14)的最优解，因此：

$$g_k^T p_k = \lambda_k^T A_k^T p_k$$

由式(6-18)，$A_k^T p_k = (a_s^T p_k) e_s$，$e_s$ 为单位矩阵的第 s 列，有：

$$g_k^T p_k = (a_s^T p_k) \lambda_k^T e_s = (\lambda_k)_s a_s^T p_k$$

由式(6-19)知，当 $(\lambda_k)_s < 0$ 时，$g_k^T p_k < 0$，从而 p_k 为下降方向。因此若$(\lambda_k)_s < 0$，就可

以找到满足式(6-18)的下降方向,特别是将问题(6-15)中的约束换成式(6-18),即删去第 s 个约束,所得到的新二次规划问题(6-15)的最优解必为一个可行的下降方向。

2. 有效集法算法流程

有效集法算法流程如下:

S1:选取初始的可行解 $x^{(1)}$,$x^{(1)}$ 满足:

$$a_i^T x^{(1)} - b_i = 0, \quad i \in E$$
$$a_i^T x^{(1)} - b_i \leqslant 0, \quad i \in I$$

求得 $x^{(1)}$ 处的有效集:

$$I(x^{(1)}) = \{a_i^T x^{(1)} - b_i = 0, i \in I\},$$

令 $k=1$。

S2:求解等式二次规划问题:

$$\min \quad q(x) = \frac{1}{2} d^T G d + g_k^T d$$
$$\text{s.t.} \quad a_i^T d = 0, \quad i \in E \cup I(x^{(k)})$$

其中:$g_k = \nabla f(x_k) = G x_k + C$,可以求得解为 d_k 。

S3:根据 d_k 性质,分两种情况:

若 $d_k = 0$,则计算相应的乘子 λ_k。若 $\lambda_k \geqslant 0$,$\forall i \in I(x^{(1)})$,则找到了最优解($x^{(k)}$ 为一般二次规划问题(6-14)的最优解,λ_k 为相应的乘子);否则求

$$\lambda_q^{(k)} = \min\{\lambda_i^{(k)} \mid i \in I(x^{(1)})\},$$

令 $x^{(k+1)} = x^{(k)}$,$I(x^{(k+1)}) = I(x^{(k)}) - \{q\}$,$k = k+1$,转 S2;

若 $d_k \neq 0$,计算 $\bar{a}_k = \min\limits_{i \notin I_k}\left\{\dfrac{b_i - a_i^T x_k}{a_i^T d_k} \mid a_i^T d_k > 0, i \notin I(x^{(k)})\right\} = \dfrac{b_t - a_t^T x_k}{a_t^T d_k}$

其中 t 为满足条件的有效约束指标,

取 $a_k = \min\{\bar{a}_k, 1\}$,令 $x^{(k+1)} = x^{(k)} + a_k d^{(k)}$ 。

如果 $a_k = \bar{a}_k$,则令 $I(x^{(k+1)}) = I(x^{(k)}) + \{t\}$,

否则令 $I(x^{(k+1)}) = I(x^{(k)})$,$k = k+1$,转 S2。

算法流程图如图 6-1 所示。

3. 二次规划的算例分析

由于有效集法的算法流程比较复杂,为了简单说明有效集法,下面给出两个算例。

算例 1:求解问题

$$\min \quad f(x) = x_1^2 + x_2^2 - 2x_1 - 4x_2$$
$$\text{s.t.} \quad c_1(x) = x_1 + x_2 - 1 \leqslant 0,$$
$$c_2(x) = -x_1 \leqslant 0,$$
$$c_3(x) = -x_2 \leqslant 0,$$

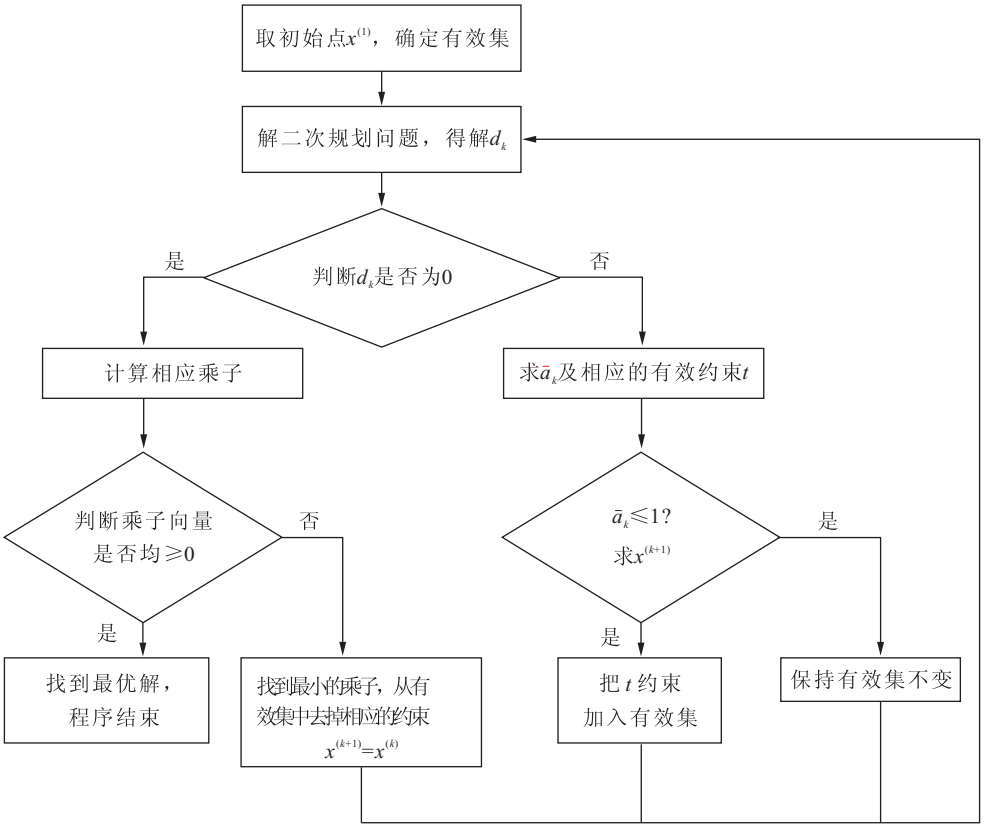


图 6-1　有效集法流程

取初始点 $x^{(1)}=(0,0)^T$ 。

解:首先,计算 $\nabla f(x)=(2x_1-2,2x_2-4)^T$。由于 $x^{(1)}=(0,0)^T$,算得初始点的有效集为 $I(x^{(1)})=\{2,3\}$,$\nabla f(x^{(1)})=(-2,-4)^T$,化成对应的等式约束问题为:

$$\begin{aligned}\min\quad & d_1^2+d_2^2-2d_1-4d_2,\\ \text{s.t.}\quad & -d_1=0,\\ & -d_2=0.\end{aligned}$$

由等式约束的一阶必要条件得到如下矩阵方程式:

$$\begin{pmatrix}2 & 0 & -1 & 0\\ 0 & 2 & 0 & -1\\ -1 & 0 & 0 & 0\\ 0 & -1 & 0 & 0\end{pmatrix}\begin{pmatrix}d_1\\ d_2\\ \lambda_2\\ \lambda_3\end{pmatrix}=\begin{pmatrix}2\\ 4\\ 0\\ 0\end{pmatrix}$$

求解方程组得到 $d_1=0, d_2=0, \lambda_2=-2, \lambda_3=-4$，因为 λ 存在小于 0 的分量，所以 $x^{(1)}=(0,0)^T$ 不是最优解。

进行第二轮迭代：取 $x^{(2)}=(0,0)^T$，则对应有效集 $I(x^{(2)})=I(x^{(1)})-\{3\}=\{2\}$，对应的二次规划问题为：

$$\begin{aligned} \min \quad & d_1^2+d_2^2-2d_1-4d_2, \\ \text{s.t.} \quad & -d_1=0, \end{aligned}$$

由等式二次规划的一阶必要条件同样得到：

$$\begin{pmatrix} 2 & 0 & -1 \\ 0 & 2 & 0 \\ -1 & 0 & 0 \end{pmatrix}\begin{pmatrix} d_1 \\ d_2 \\ \lambda_2 \end{pmatrix}=\begin{pmatrix} 2 \\ 4 \\ 0 \end{pmatrix}$$

求解方程组可以得到 $d_1=0, d_2=2$，即 $d^{(2)}=(0,2)^T$，因为 d 不全为 0，所以计算：

$$\bar{\alpha}_2=\frac{b_1-a_1^T x^{(2)}}{a_1^T d^{(2)}}=\frac{1-0}{2}=\frac{1}{2},$$

$$\alpha_2=\min\left\{\frac{1}{2},1\right\}=\frac{1}{2},$$

所以

$$x^{(3)}=x^{(2)}+\alpha_2 d^{(2)}=(0,1)^T,$$

由于 $\bar{\alpha}_2=\alpha_2$，因此在 $x^{(3)}$ 处的有效集要加入第一个约束，有效集变为：

$$I(x^{(3)})=\{1,2\}.$$

第三次迭代：在 $x^{(3)}$ 处 $\nabla f(x^{(3)})=(-2,-2)^T$，对应的二次规划问题为：

$$\begin{aligned} \min \quad & d_1^2+d_2^2-2d_1-2d_2, \\ \text{s.t.} \quad & d_1+d_2=0, \\ & -d_1=0. \end{aligned}$$

由等式二次规划的一阶必要条件得到：

$$\begin{pmatrix} 2 & 0 & 1 & -1 \\ 0 & 2 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \end{pmatrix}\begin{pmatrix} d_1 \\ d_2 \\ \lambda_1 \\ \lambda_2 \end{pmatrix}=\begin{pmatrix} 2 \\ 2 \\ 0 \\ 0 \end{pmatrix}$$

解方程得 $d_1=0, d_2=0, \lambda_1=2, \lambda_2=2$，并且其拉格朗日乘子向量的所有分量均大于 0，根据定理可得，$x^{(3)}=(0,1)^T$ 就是问题的最优解，$\lambda^{(3)}=(2,0,0)^T$ 是其相应的拉格朗日乘子。

程序运行结果如图 6－2 所示。

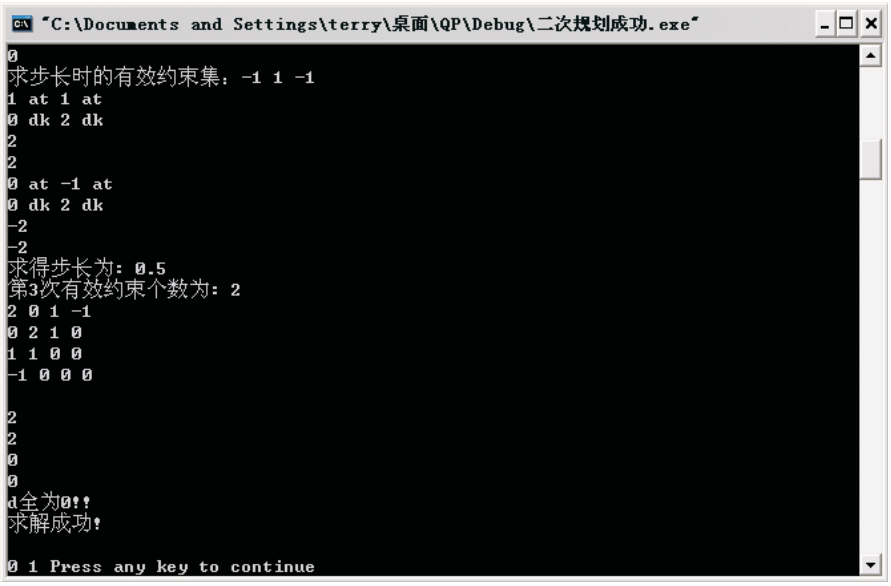


图 6-2　算例 1 的程序运行结果

算例 2:求解问题

$$
\begin{aligned}
\min \quad & f(x) = (x_1 - 1)^2 + (x_2 - 2.5)^2 \\
\text{s.t.} \quad & c_1(x) = -x_1 + 2x_2 - 2 \leqslant 0, \\
& c_2(x) = x_1 + 2x_2 - 6 \leqslant 0, \\
& c_3(x) = x_1 - 2x_2 - 2 \leqslant 0, \\
& c_4(x) = -x_1 \leqslant 0, \\
& c_5(x) = -x_2 \leqslant 0.
\end{aligned}
$$

解:第一轮迭代:选取初始可行解 $x^{(0)} = (2,0)^T$,求得其有效集 $I(x^{(0)}) = \{3,5\}$,从一阶必要条件求解方程组得到其乘子为$(\lambda_3,\lambda_5) = (-2,-1)^T$,$d$ 全为 0。因为 λ_3 最小且小于 0,所以有效集个数减一个,即 $I(x^{(1)}) = \{5\}$。令 $k = 1$,$x^{(1)} = (2,0)^T$,再求解子问题得到 $d_1 = (-1,0)^T$,则步长 $\alpha_1 = 1$,$x^{(2)} = (1,0)^T$,其有效集为:$I(x^{(2)}) = \{5\}$ 。

第二轮迭代:求解子问题得 d 全为 0,$\lambda_5 = -5$,有效约束再减一个,则 $I(x^{(3)}) = \varphi$ 。

第三轮迭代:求解无约束二次规划问题,解方程组求得最优解 $d_3 = (0,2.5)^T$,$\alpha_1 = 0.6$,求得 $x^{(4)} = (1,1.5)^T$,加上一个约束 $I(x^{(4)}) = \{1\}$ 。

同理求得 $d_4 = (0.4,0.2)^T$,$I(x^{(5)}) = \{1\}$,$x^{(5)} = (1.4,1.7)^T$。

第四轮迭代:求得 d 为 0,且相应的拉格朗日乘子向量的所有分量均大于 0,因此找到了最优解 $x = (1.4,1.7)^T$ 。

程序运行结果如图 6－3 所示。

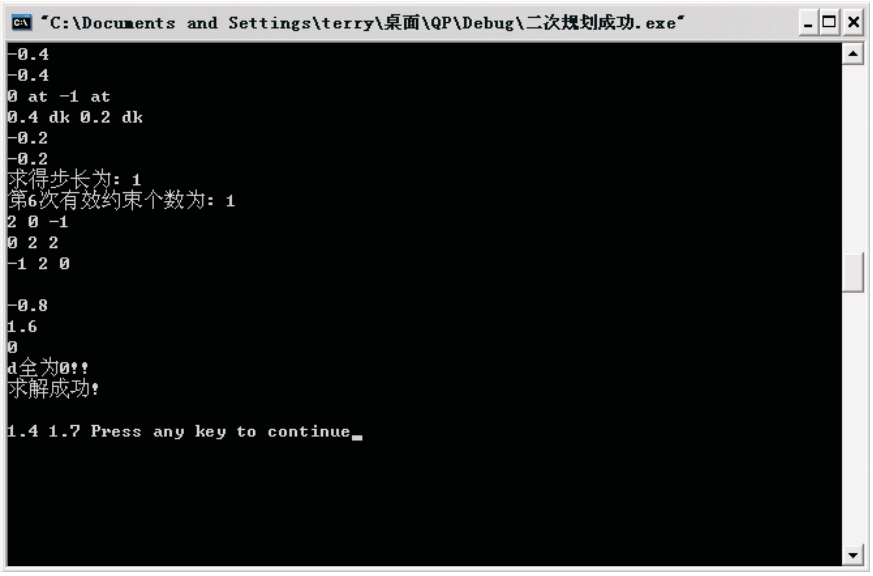


图 6－3　算例 2 的程序运行结果

在上例中，若把第一个条件的第一个未知数的符号改变，则得出的解就是(1,2.5)，这是很显然的，同时进一步说明了算法的正确性。

6.1.4　SQP 方法

前面已经实现了 SQP 的核心算法 QP，下面介绍 SQP 算法。

1. 最优化方法的一般迭代思想

最优化方法的一般迭代思想可概括如下：

S1：给定一个最优点 $x^{(0)}$，令 $k=0$；

S2：如果 $x^{(k)}$ 满足最优解收敛条件，则停止迭代；

S3：确定 $x^{(k)}$ 一个改善的修正量 $\xi^{(k)}$；

S4：得到一个更好的猜测点 $x^{(k+1)}=x^{(k)}+\xi^{(k)}$，令 $k=k+1$，转 S2；

2. SQP 方法的算法流程

SQP 的一般迭代格式(Storn，*et al*，1995)可以描述为：

S1：选取初始点 $x^{(1)}$，正定的 n 维矩阵 B_1，令 $k=1$；

S2：求解二次规划子问题 $Q(x^{(k)},B_k)$：

$$\begin{aligned}
\min \quad & \frac{1}{2}d^T B_k d + d^T g^{(k)} \\
\text{s.t.} \quad & c_i(x^{(k)}) + d^T \nabla c_i(x^{(k)}) = 0, \quad i \in E = \{1,2,\cdots,m_e\}, \\
& c_i(x^{(k)}) + d^T \nabla c_i(x^{(k)}) \geqslant 0, \quad i \in I = \{m_e, m_e+1,\cdots,m\},
\end{aligned}$$

得到其解 $d^{(k)}$ 和其对应的拉格朗日乘子 $\lambda^{(k+1)}$ 。

S3：针对所选的效益函数从点 $x^{(k)}$ 沿方向 $d^{(k)}$ 进行线性搜索，确定步长 a_k，并令 $x^{(k+1)} = x^{(k)} + a_k d^{(k)}$ ；

S4：利用 $\lambda^{(k+1)}$ 等信息，采用 BFGS 拟牛顿修正公式（Kennedy，*et al*，1995）对 B_k 进行修正，以得到对 $w = \nabla^2 L$ 的新近似 B_{k+1}，令 $k = k+1$，转 S2。

3. SQP 算法的算例

例：考虑非线性优化问题

$$\begin{aligned}
\min \quad & f(x) = e^{3x_1+4x_2} \\
\text{s.t.} \quad & c_1(x) = x_1^2 + x_2^2 = 1
\end{aligned}$$

以点 $x_0 = (-0.7, -0.7)$ 为初始点，逐次求解下列二次规划子问题：

$$\begin{aligned}
\min \quad & q(d) = \frac{1}{2}d^T(G(x_k) - \lambda_k \nabla^2 c_1(x_k))d + g_k^T d \\
\text{s.t.} \quad & \nabla c_1(x_k)^T d + c_1(x_k) = 0
\end{aligned}$$

的最优解和拉格朗日乘子，其中：

$$g(x) = e^{3x_1+4x_2}\begin{pmatrix}3\\4\end{pmatrix}, \qquad \nabla c_1(x) = 2\begin{pmatrix}x_1\\x_2\end{pmatrix},$$

$$G(x) = e^{3x_1+4x_2}\begin{pmatrix}9 & 12\\12 & 16\end{pmatrix}, \quad \nabla^2 c_1(x) = 2\begin{pmatrix}2 & 0\\0 & 2\end{pmatrix}$$

在初始点求得第一次二次规划解为：

$$d_1 = \begin{pmatrix}0.141\,96\\-0.156\,24\end{pmatrix}, \lambda_1^{(1)} = -0.014\,808$$

由此得到：

$$x^{(1)} = \begin{pmatrix}-0.7\\-0.7\end{pmatrix} + \begin{pmatrix}0.141\,96\\-0.156\,24\end{pmatrix} = \begin{pmatrix}-0.558\,04\\-0.856\,24\end{pmatrix}$$

经过五次迭代，得到最优解：

$x^* = (-0.6, -0.8)^T$，其对应的拉格朗日乘子为 $\lambda_1^{(5)} = -0.016\,845$。

§6.2　改进的 SQP 算法

对于复杂的非线性搜索问题，遗传算法的搜索不依赖于目标函数的梯度信息，也不需要

初始值信息，有着全局搜索能力强的优点，但存在着求解精确度低、收敛慢等问题。而基本的 SQP 方法虽然求解速度快、精确度高，有着极强的局部收敛性，但又存在着初始点敏感等问题。因此，如何构造一个新的算法，它能结合遗传算法和 SQP 算法的优点呢？下面从两个方面介绍这方面的工作。

6.2.1 全局-局部结合搜索

全局-局部搜索是一种比较简单的结合思想，是将全局算法和局部算法结合使用。本书中全局-局部搜索思想主要是为了克服 SQP 的初始点敏感问题，而且不失精确度和收敛速度。主要思想就是将遗传算法搜索得到的优化解作为 SQP 方法的初始点。哈尔滨工业大学(张然等，2008)也曾运用这种思想求解小天体交会问题。

全局-局部协同搜索方式步骤如下：

S1：根据任务要求选择适当的发射窗口和转移时间作为搜索域；

S2：选择遗传算法的搜索参数，并设置初始搜索的终止条件；

S3：采用遗传算法得到的搜索结果，作为 SQP 搜索的初始条件；

S4：用 SQP 方法进行局部搜索，得到精确解。

该方法的流程如图 6－4 所示，图 6－5 给出了搜索过程的简单示意图。

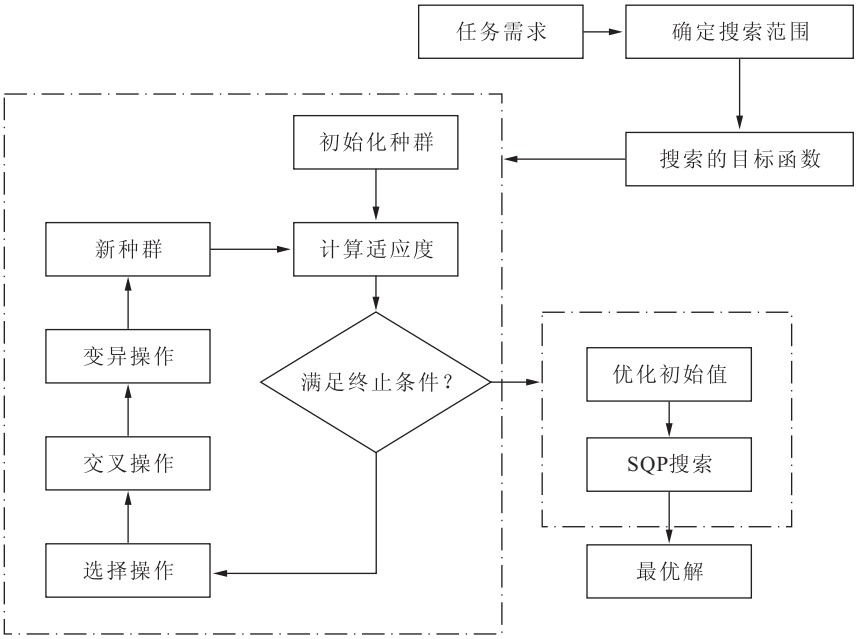


图 6－4 全局-局部方法流程

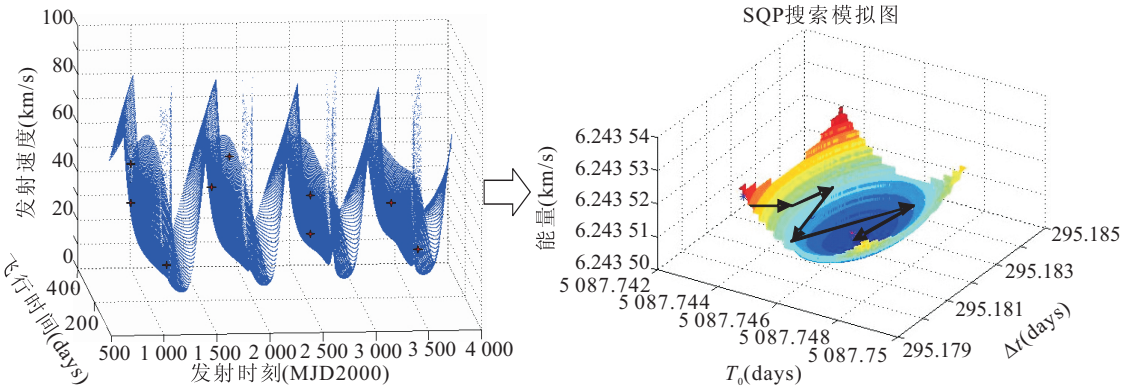


图 6－5　全局-局部搜索示意图

6.2.2　基于 SQP 算子的混合遗传算法

基于 SQP 的混合遗传算法思想主要是，为了克服遗传算法的精确度低、收敛慢等问题，同时也可以避免 SQP 的局部收敛性。主要思想也就是把 SQP 方法作为遗传算法的变异策略，利用 SQP 始终是朝着目标函数下降方向迭代的性质和其精确特性，让遗传算法的变异更有启发性。其流程如图 6－6 所示。

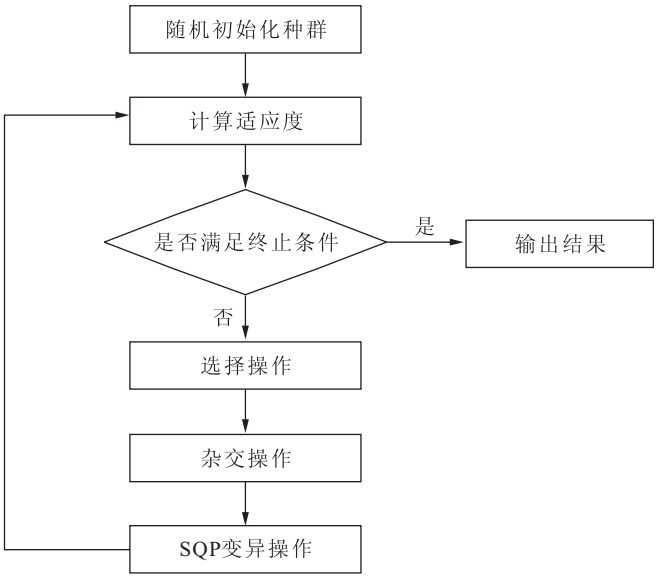


图 6－6　基于 SQP 变异算子的混合遗传算法流程

§6.3 实验结果

在本节里，将采用基本的 SQP 方法、GA、全局-局部方法以及混合遗传算法 4 种算法分别对式(4-11)进行求解，将求得的结果与文献的结果进行比较的同时，还将从求解时间、精确度以及改进效果等方面对这 4 种算法来进行比较。

6.3.1 遗传算法设计

1. 种群的初始化

$$t_0^j = (t_{0\max} - t_{0\min}) \cdot b + t_{0\min}$$

$$\Delta T^j = (\Delta T_{\max} - \Delta T_{\min}) \cdot b + \Delta T_{\min}$$

式中：$j = 1,2,\cdots,M$，M 为群体规模；$T^j = (t_0^j, \Delta T^j)$ 是第 j 组个体，分别为发射时间、交会的飞行时间；$b \in [0,1]$。

2. 适应度的求法

由于这里要求的是最小值，所以 f 越小，这个个体的适应度就越大。于是采用下式计算：

$$f^j = C - f(t_0^j, \Delta T^j)$$

式中：C 为常数，其值应该大于或等于 $\max(f(t_0^j, \Delta T^j))$。

3. 选择算子的设计

个体被选择的概率 $p^j = f^j / F$；

f^j 为个体的适应度，F 为种群的总适应度，即个体适应度值总和。个体适应度越高，被选中的几率就越大。

4. 杂交算子的设计

$$T_1^j = T^{j1} - b \cdot (T^{j1} - T^{j2})$$

$$T_2^j = T^{j2} + b \cdot (T^{j1} - T^{j2})$$

其中，b 为 0 ～ 1 间的随机数，若 T_1^j 和 T_2^j 比原来个体的适应度高，则选择。杂交概率为 p_c，杂交个体的数目为 $M \cdot p_c$。

5. 变异算子的设计

假设第 g 代的个体为 T_g，则新个体

$$T_{g+1} = \begin{cases} T_g + \Delta(g,e), & R = 0 \\ T_g - \Delta(g,e), & R = 1 \end{cases}$$

其中 $e = T_{\max} - T_{\min}$，$\Delta(g,e) = e(1 - r^{(1-\frac{g}{G})^b})$，$r \in [0,1]$，$R \in \{0,1\}$，$G$ 为最大允许迭代次数，b 为正整数，g 为当前迭代次数。这样的好处是迭代后期方便收敛。

这些是下面 GA 使用的算子，需要说明的是，在全局-局部策略中的 GA 模块中也采用上述算子。在基于 SQP 的混合遗传算法中，采用 SQP 算子作为变异算子，其余算子相同，这样做才能更加公平地比较 4 种方法的优劣。

6.3.2　实验数据及结果

1. 实验一

Myatt(2003)给出了地火最优转移轨道求解问题在发射时刻 $t_0 \in [800, 3\ 800]$MJD2000(即 2003 年 3 月到 2011 年 5 月)和转移时长 $\Delta t \in [100, 400]$(days)，为了与 ESA 的文献结果相比，本书首先采用此数据做实验。可以看到目标函数图像如图 6－7 所示。

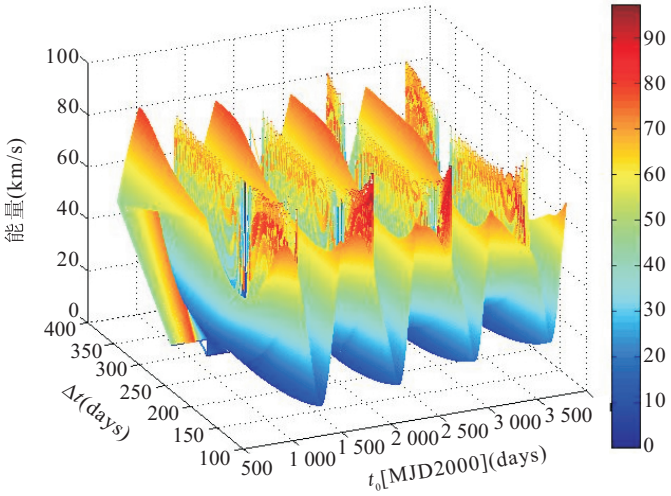


图 6－7　目标函数在解空间 $t_0 \in [800, 3\ 800]$ MJD2000、$\Delta t \in [100, 400]$Days 的图像

(1)基本的 SQP 方法。

基本的 SQP 方法得到的结果如表 6－1 所示。

解在空间中的分布情况如图 6－9 所示。

从 SQP 算法的优化结果可以看出，找到了文献(Myatt，*et al*，2003)给出的全部 11 个局部最优解，全局最优解是第一个解，优化结果与文献(Myatt，*et al*，2003)基本一致，两者之间的差别可能来源于星历计算或其他一些参数的取值精度。

(2)全局-局部策略。

以迭代次数为 50 次作为初始解产生的终止条件，优化的初始点如表 6－2 所示，这里只列出前 6 个初始点。

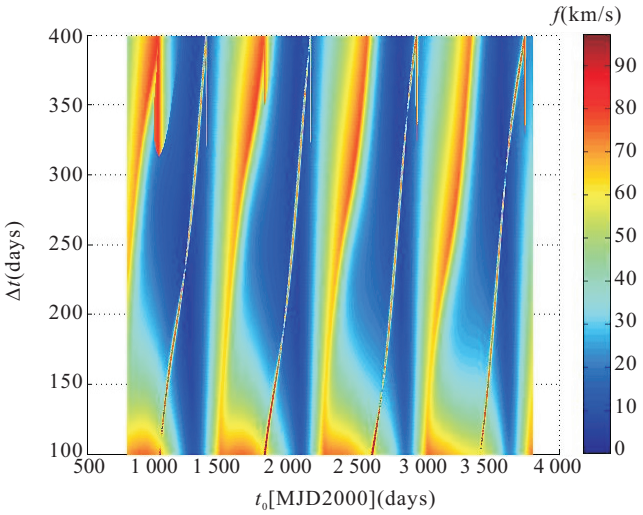


图 6－8　目标函数在解空间 $t_0 \in [800, 3\ 800]$ MJD2000、$\Delta t \in [100, 400]$的投影

表 6－1　SQP 算法的优化结果与文献(马文臻,2006)对比

SQP	SQP 算法的优化结果			文献(马文臻,2006)		
	t_0(MJD2000)	Δt(days)	f(km/s)	t_0(MJD2000)	Δt	f(km/s)
1	1 253.326	203.755	5.668 619 496	1 253.7	202.9	5.667
2	3 573.151	324.148	5.669 498 730	3 573.4	323.2	5.673
3	2 812.504	345.611	6.185 062 109	2 812.5	344.2	6.223
4	1 226.601	221.756	6.238 280 388	1 225.5	233.2	6.353
5	2 057.305	216.541	6.789 894 329	2 057.5	216.5	6.835
6	3 597.01	309.876	6.452 845 676	3 597.8	273.5	6.861
7	2 048.2	352.582	7.077 496 582	2 048.2	350.5	7.106
8	2 834.01	356.717	6.713 031 952	2 834.3	246.0	7.369
9	3 766.779	400.0	21.545 820 101	3 766.8	400.0	21.633
10	798.112	400.0	44.063 975 138	800.0	400.0	44.292
11	3 799. 663	400.0	23.908 685 032	3 800.0	126.7	63.289

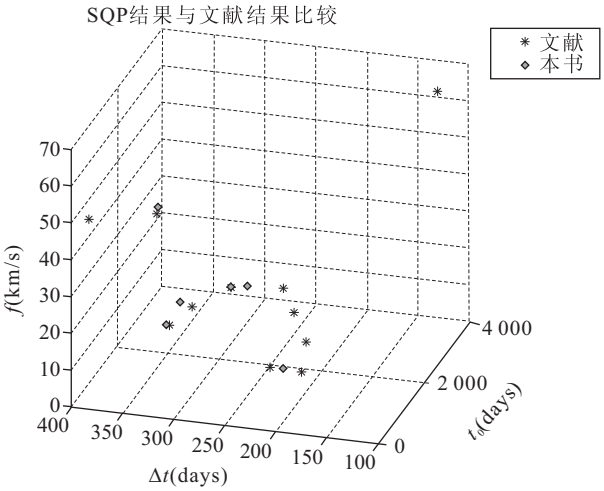


图 6－9　解的分布情况

表 6－2　全局初始点结果

全局-局部	t_0(MJD2000)	Δt (days)	$\Delta v_1+\Delta v_2$(km/s)
1	1 252.31	205.416	5.670 415 461
2	1 267.17	205.796	5.886 025 221
3	2 834.78	223.561	7.693 348 570
4	2 056.39	201.866	6.903 607 110
5	2 052.48	226.711	6.964 716 078
6	3 569.73	335.33	5.726 666 791
……	……	……	……

用 SQP 精确搜索后得到结果对应表 6－3 中的初始点。

表 6－3　局部搜索点结果

全局-局部	t_0(MJD2000)	Δt (days)	$\Delta v_1+\Delta v_2$(km/s)
1	1 253.334	203.754	5.668 619 550
2	1 253.327	203.760	5.668 619 504
3	2 834.01	356.717	6.713 031 952
4	2 057.303	216.539	6.789 887 339
5	2 057.309	216.536	6.789 895 843
6	3 573.149	324.152	5.669 498 752
……	……	……	……

挑选出与文献(Myatt,*et al*,2003)中接近的点如表 6－4 所示。

表 6－4 全局-局部策略与 GA 的结果比较

min	GA			全局-局部策略		
	t_0 (MJD2000)	Δt (days)	f (km/s)	t_0 (MJD2000)	Δt (days)	f (km/s)
1	1 253.33	203.758	5.668 619 513	1 253.327	203.760	5.668 619 504
2	3 573.26	323.801	5.669 567 687	3 573.149	324.152	5.669 498 752
3	2 812.27	346.679	6.185 774 091	2 812.538	345.577	6.185 064 853
4	2 056.51	215.03	6.788 223 229	2 057.303	216.539	6.789 887 339
5	3 597.19	248.369	7.267 914 925	3 597.003	309.873	6.452 394 064
6				2 834.01	356.717	6.713 031 952
运算时间(s)	6.332			1.604＋0.437＝2.041		

表 6－4 中 GA 的结果为实验 20 次的结果,由于适应度的选择策略原因,全局-局部策略不能找到所有的局部最优解。实验结果证明,全局-局部策略可以有效地改进 SQP 初始点敏感问题,而且比 GA 更精确,耗时仅为 GA 的 32.2%,但不能找到所有的局部最优解。

(3)SGA 搜索 。

本例中,SGA 的变异策略为每演化 10 代挑选其中适应度从优到劣排在前 15%的个体进行随机 SQP 变异。做了 20 次实验,种群规模为 100,SGA 的结果如表 6－5 所示,这里只列举前 5 个。

表 6－5 SGA 的结果

SGA	t_0(MJD2000)	Δt (days)	$\Delta v_1+\Delta v_2$(km/s)
1	1 253.33	203.754	5.668 619 507
2	2 056.87	217.009	6.790 235 534
3	1 253.23	204.143	5.668 694 913
4	2 056.56	215.268	6.788 188 683
5	2 057.64	215.81	6.790 274 368
……	……	……	……

挑选出与文献(Myatt,*et al*,2003)中接近的点,GA 与 SGA 的比较如表 6－6 所示。

表 6－6　SGA 与 GA 的结果比较

min	GA			SGA		
	t_0 (MJD2000)	Δt (days)	f (km/s)	t_0 (MJD2000)	Δt (days)	f (km/s)
1	1 253.33	203.758	5.668 619 513	1 253.33	203.754	5.668 619 507
2	3 573.26	323.801	5.669 567 687	3 573.08	323.846	5.669 549 525
3	2 812.27	346.679	6.185 774 091	2 812.58	345.65	6.185 068 974
4	2 056.51	215.03	6.788 223 229	2 056.56	215.268	6.788 188 683
5	3 597.19	248.369	7.267 914 925			
运算时间(s)	6.332			5.257		

由于适应度的选择策略原因，SGA 也不能找到所有的局部最优解。从表 6－6 中可以看到，求解结果与表 6－1 中的基本 SQP 求解结果非常接近，而且算法具有全局搜索性能。实验结果证明，SGA 算法比 GA 收敛得更快，耗时为 GA 的 83.02%，精确度更高。但是同样不能找到所有的局部最优解。

3 种算法的比较如表 6－7 所示，这里只比较全局最优解，由于其他的局部极小点只有基本 SQP 算法才能够全部得到，因此这里不作比较，全局最优解为发射窗口 2003 年 6 月左右、交会时间为 203.755 天时，能量最省。

GA 和 SGA 的 20 次求解平均值以及方差如表 6－8 所示。

表 6－7　实验一的 3 种算法的结果比较

	基本 SQP 方法	GA	全局-局部策略	SGA
t_0(MJD2000)(days)	1 253.326	1 253.33	1 253.327	1 253.33
ΔT(days)	203.755	203.758	203.760	203.754
f(km/s)	5.668 619 496	5.668 619 513	5.668 619 504	5.668 619 507
计算时间(s)	0.437(单次)	6.332	2.041	5.257

表 6－8　GA 与 SGA 在 20 次求解后的统计结果

	实验次数	最大值	最小值	平均值	方差
GA	20	5.668 299 80	5.668 195 13	5.668 217 70	1.1×10^{-8}
SGA	20	5.668 304 75	5.668 195 07	5.668 213 10	1.3×10^{-8}

基本的 SQP 算法在时间上与其他 3 种算法比较并没有任何意义，因为 SQP 存在初始点敏感的问题，每次求解初始点不同，得到的结果也不同。假设用 SQP 方法在一次随机选取初始点的情况下求得全局最优解的概率为 p^*，那么 n 次后求解均陷入局部最优的概率为：

$$p_{\text{fail}} = (1 - p^*)^n$$

那么需要随机求解次数与陷入局部最优的概率的关系为：

$$n = \frac{\log p_{\text{fail}}}{\log(1 - p^*)}$$

经过简单的计算可以求出，至少经过 16 次的随机试验，才能使求得全局最优解的概率达到 99%。因此，SQP 的求解时间是不定的，但是基本的 SQP 可以找到更多的局部最优点，这对于工程上的分析是有意义的。

这个问题可以考虑引用概率算法的思想。常用的概率算法有数值概率算法、Sherwood 算法、Las Vegas 算法以及 Monte Carlo 算法等。数值概率算法一般是用来求解一些问题的近似解，且求解时间越长，结果越精确；Sherwood 算法是用来消除确定性算法在最坏情况下与平均情况下的计算复杂性有较大差别时，最坏情况与特定实例间的关联性；Las Vegas 算法和 Monte Carlo 算法的特点是，若反复用一个算法求解多次，那么求得正确解的概率任意高，但是 Monte Carlo 算法有时候求得的解未必正确。

2. 实验二

发射时刻 $t_0 \in [3\ 800, 6\ 000]$ MJD2000（即 2011 年 5 月到 2017 年 6 月）和转移时长 $\Delta t \in [100, 400]$Days 的双脉冲转移能量函数如图 6－10。

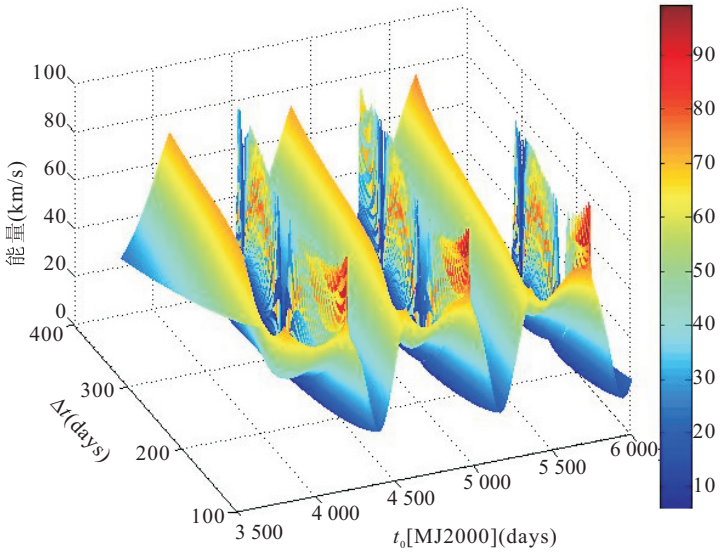


图 6－10　目标函数在解空间 $t_0 \in [3\ 800, 6\ 000]$ MJD2000、$\Delta t \in [100, 400]$Days 的图像

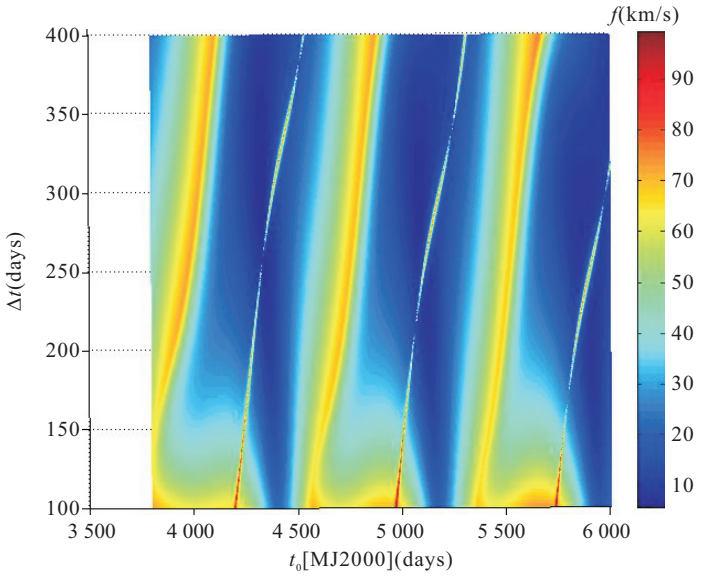


图 6－11　目标函数在解空间 $t_0 \in [3\ 800, 6\ 000]$ MJD2000、$\Delta t \in [100, 400]$Days 的投影图

这里实验过程与实验一类似，这里给出 4 种算法求得的全局最优解的比较结果。如表 6－9 所示。

表 6－9　实验二的 3 种算法的结果比较

	基本 SQP 方法	GA	全局-局部策略	SGA
t_0[MJD2000](days)	4 329.758	4 329.75	4 329.753	4 329.76
t(days)	306.594	306.589	306.597	306.596
f(km/s)	5.698 484 072	5.698 484 176	5.698 484 086	5.698 484 091
计算时间(s)	0.422(单次)	6.347	2.143	5.222

全局最优解为发射窗口 2012 年 11 月左右、交会时间为 306.594 天时，能量最省。可以看到，全局-局部策略的搜索时间为 GA 的 33.76%；SGA 的搜索时间为 GA 的 82.28%。

6.3.3　算法比较结论

从上一节中列出的各种算法的比较结果中可以看到，全局-局部策略的方法在求解地球-火星双脉冲最优转移轨道中的表现是最出色的，不仅耗时最少，而且精确度高，全局策略为 SQP 搜索提供了优秀的初始点，从而克服了 SQP 对初始点敏感的问题，而 SQP 极强的局部

搜索能力，为算法本身保证了求解效率和精确度；其次，SGA 算法的表现也不错，在耗时和精确度方面同样比 GA 要优，这是因为其采用 SQP 变异这种启发式的变异策略，从而保证算法在 150 代以内便可以迅速收敛，而且精确度也很高，但是从表 6－8 中的方差项可以看到 SGA 比 GA 稳定性稍微差一点，但是基本上可以忽略。若需要找到多个点，SGA 比全局-局部策略要好，因为全局-局部策略需要针对每个点来进行 SQP 搜索。GA 算法可以解决这个问题，但是需要在 200 代左右才能收敛，在耗时和精确度上表现不如其他算法。SQP 单次的运算效率是最高的，但是由于它对初始点敏感，因此耗时是一个不确定因素，当决策变量的维数增多时，其初始点的选择是异常麻烦的。

需要说明的是，由于算法策略原因，只有 SQP 方法能够找到所有 11 个局部极小点，而全局-局部结合搜索策略、SGA 和 GA 只能找到包括全局极小点在内的部分局部极小点。

第七章　行星际轨道优化设计的单纯形算法

单纯形算法产生于20世纪60年代直接搜索算法发展的黄金时期。它的基本思想是由Spendley等于1962年在论文中首次提出的。之后针对在理论和实践中遇到的问题，又提出了第二代和第三代单纯形算法，使该算法得到进一步的发展和完善，表现出更好的性能(Fred，*et al*，1991)。目前单纯形算法在工程与经济等领域已得到广泛的应用。

§7.1　单纯形算法

单纯形算法是求解无约束非线性规划问题的一种直接搜索算法，相对于以上需要计算导数的算法，直接搜索方法只要目标函数是连续的，无需计算目标函数的偏导数，甚至不要求偏导数一定存在，只需要计算目标函数值的值就够了。这类算法思想比较简单，编制程序也比较容易，对于变量不多的问题，能够收到较好的效果(陈宝林，1989)。

单纯形算法是建立在单纯形的概念基础上的。几何学上，单纯形或者 n -单纯形是和三角形类似的 n 维几何体。

例如，0 -单纯形就是点，1 -单纯形就是线段，2 -单纯形就是三角形，3 -单纯形就是四面体，而 4 -单纯形是一个五胞体(每种情况都包含内部)。

任何 $n+1$ 个点集的非空子集的凸包定义了一个 n -单纯形，称为该$(n+1)$-单纯形的面。面本身也是单纯形。$(n+1)$的 $m+1$ 子集的凸包是一个 m -单纯形，称为 n -单纯形的 m -面。0 -面(也即一个点构成的面)称为顶点，而 1 -面称为边，$(n-1)$-面称为面片，而 n -面就是 n -单纯形本身。

7.1.1　早期单纯形算法

单纯形具有这样一个特性，如果用某个点的相对于单纯形中心点的对称点代替该点后仍然是单纯形(图 7 - 1)。因为在搜索最优解时需要反射一个顶点，根据这一特征，反射之后仍然构成一个单纯形，简化了过程。1962 年 Spendley 等在论文中提出了单纯形算法的雏形。

在初始单纯形构造好后，首先确定这个单纯形中的“最差”顶点，之后找出除这个点之外的其他所有点的几何中心点，并以这个中心点为中心反射“最差”顶点并替代之，形成新的单

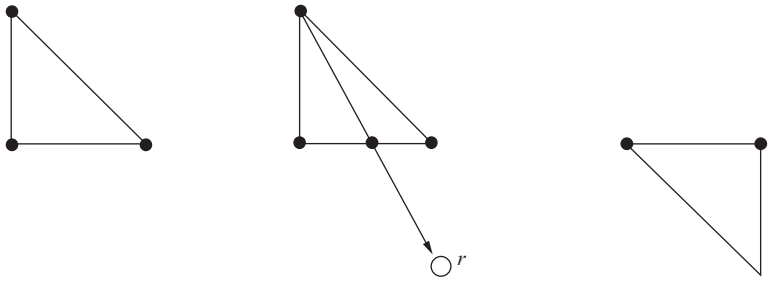


图 7－1　单纯形镜像过程

纯形。在这个过程中单纯形移动是镜像的。但是如果映射后的点仍然是最差的，则选择“次差”顶点继续这个过程，最终的目标是取代“最好”顶点或者确定出最优顶点是一个最优解的候选。直到那时，算法一直在通过相对面的中心翻转顶点来移动单纯形（而不是最好顶点）。

对于这种最初的单纯形算法，反射是它的核心迭代策略，但是这种过于简单的算法思想甚至在理论上就存在着缺陷。每一次的反射都是通过相对面的中心翻转来镜像移动单纯形，但是这个翻转方向是否就是可靠的寻找更优点的期望方向则难以保证。

在最极端的情况下，这种镜像的移动甚至会造成单纯形的循环序列，致使一系列的迭代后又返回到出发点的尴尬局面。对于这个缺陷的论述，Spendley 等（1962）在论文中已经给出。图 7－2 为 Spendley 用二维演示的一个例子，这里 5 步映射就回到了开始的地方，没有取代 x_k。

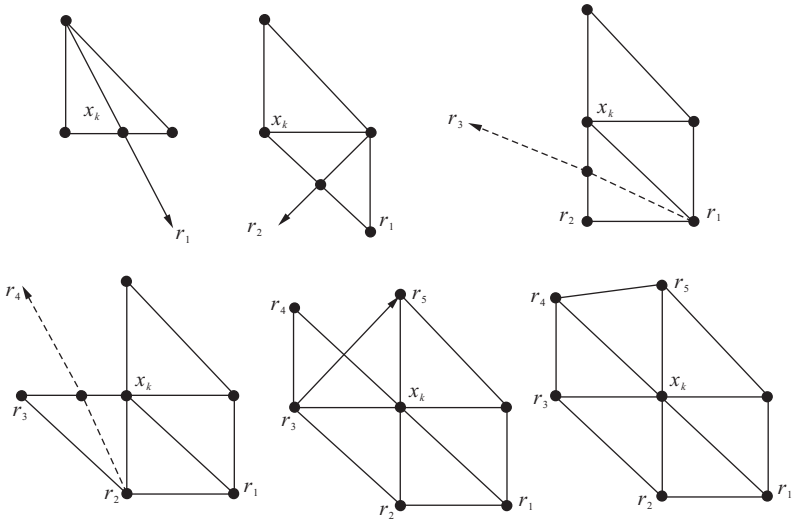


图 7－2　镜像循环

7.1.2　Nelder－Mead 单纯形算法

为了克服 Spendley 单纯形算法的缺陷，Nelder 和 Mead 在 1965 年发表的论文中提出了一些针对性的改进，这是迄今最为流行的单纯形算法。在这篇论文中，给出了一个启发性的公式，用来确定什么时候单纯形已绕最好点足够长而可以确定最优点的范围。当出现这种情况时，有两种解决方法：延长或缩短连接到最好顶点的边长，或者重新用获得快速局部收敛的更高阶的方法。

Nelder－Mead 的算法是为了能够更好适应目标函数特性，并提出了两点改进：一是在原来的基本操作-反射的基础上，提出了在反射成功时加强反射效果而增加延伸策略，在反射失败时削减反射的效果而增加压缩策略；二是当怀疑原来镜像移动的中心是否能够更为接近最优候选解时，修改原来单纯形的形状。

尽管 Nelder－Mead 单纯形算法中仍然存在着一些缺陷，但是它的综合效能是比较理想的。在所有的直接搜索法中，Nelder－Mead 单纯形算法是在数值软件包中最常见的。Nelder 和 Mead 的原始论文也是最常用的科学索引文献。

Nelder－Mead 单纯形算法的算法流程图见图 7－3，算法思想如下：

(1)构造初始单纯形。选取 $x^{(0)}$ 为起点，取一正数 h 为步长，令 $x^{(i)}=x^{(0)}+he^{(i)}$，其中 $e^{(i)}$ 为第 i 个单位坐标向量 $i=1,2,\cdots,n$。

计算各顶点的函数值 $f(x)$：$f_i=f(x^{(i)})$，$i=0,1,\cdots,n$。然后比较诸函数值的大小，选出最小值点 $x^{(L)}$、最大值点 $x^{(H)}$（对目标函数求极小而言）及次大值点 $x^{(G)}$，并计算其函数值 $f_L=f(x^{(L)})$、$f_H=f(x^{(H)})$、$f_G=f(x^{(G)})$。

(2)求反射点 $x^{(R)}$。计算除去最大值点 $x^{(H)}$ 以外的 n 个点的形心 $x^{(C)}$，再进行反射，得到反射点 $x^{(R)}=x^{(C)}+\alpha(x^{(C)}-x^{(H)})$，其中 α 为反射系数，一般取 $\alpha=1$。

反射后比较 f_R 与 f_L、f_G 有三种情形，如果 $f_R<f_L$，则转入步骤(3)；如果 $f_R>f_G$，则转入步骤(5)；否则转入步骤(4)。

(3) 延伸。若 $f_R<f_L$，则进行延伸，即令 $x^{(E)}=x^{(C)}+\beta(x^{(R)}-x^{(C)})$，其中 β 为延伸系数。计算延伸点函数值 f_E，若 $f_E<f_R$，则以点 $x^{(S)}$ 取代 $x^{(H)}$，构成新的单纯形，转步骤(7)；否则以点 $x^{(R)}$ 取代 $x^{(H)}$，转步骤(7)。

(4) 直接替换。若 $f_L\leqslant f_R\leqslant f_G$，则以点 $x^{(R)}$ 取代 $x^{(H)}$，转步骤(7)。

(5) 压缩。若 $f_R>f_G$，令 $f(x^{(H')})=\min\{f(x^{(H)}),f(x^{(R)})\}$。进行压缩：$x^{(S)}=x^{(C)}+\gamma(x^{(H')}-x^{(C)})$，得到压缩点 $x^{(S)}$。计算 $f_S=x^{(S)}$，如 $f_S\leqslant f_{H'}$，则压缩成功，转步骤(7)；否则转步骤(6)。

(6) 收缩。调整单纯形的形状：令，$x^{(i)}=x^{(i)}+\frac{1}{2}(x^{(L)}-x^{(i)})$，其中 $i=0,1,\cdots,n$。转入步骤(7)。

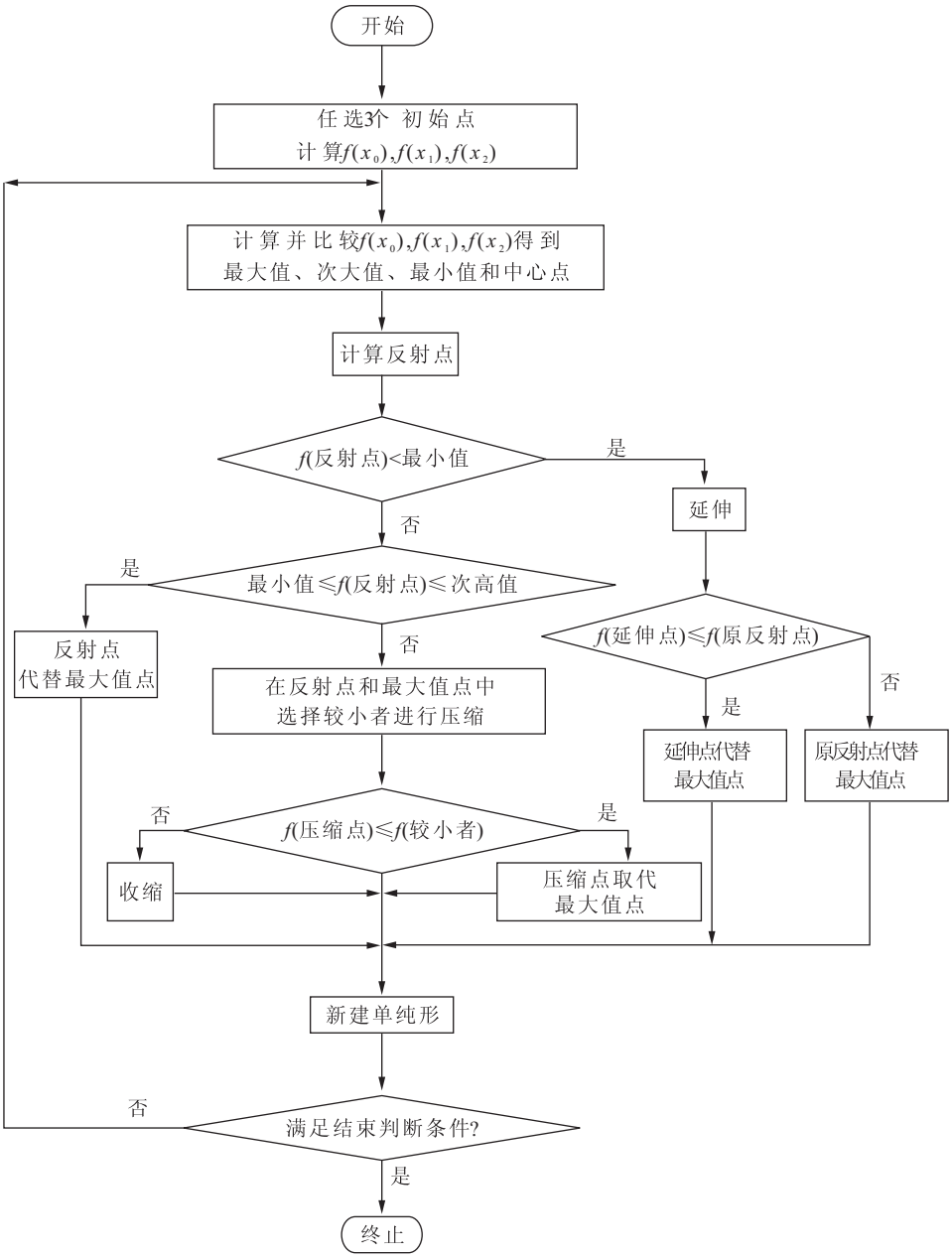


图 7－3　Nelder－Mead 算法流程图

(7) 检验。是否满足收敛准则：$\{\frac{1}{n+1}\sum_{i=1}^{n+1}[f(x_i)-f(\dot{x})]^2\}<\varepsilon$，满足则结束，否则转步骤(2)。

7.1.3　加权形心单纯形法

早期单纯形和 Nelder－Mead 单纯形都是采用去掉点的反射方向为新试验点的搜索方向，这就意味着，去掉点的反射方向作为近似的优化方向，就是梯度变化最大的方向。实际上，这个方向是一个近似的梯度最大方向，这样的搜索结果可能导致搜索次数的增加和搜索结果精度的降低。为了解决这个问题，Routh 提出了加权形心法，加权形心法利用加权形心代替单纯的反射形心，使新点的搜索方向更接近实际的最优方向。

如图 7－4，使 ω、B、C 三个顶点组成的一个二因素的优化过程的一个单纯形，并知 C 点的响应最坏，B 的响应最好。如果搜索优化过程中函数不出现异常，那么搜索最优点的方向明显应当更靠近 ωB 的方向，而不是靠近 ωC 的方向。因此可以通过加权的办法来使搜索的方向由原来的 ωE（反射方向）变为 $\omega E'$ 方向（加权方向），此时用加权形心点 O_ω 代替反射形心点 O。

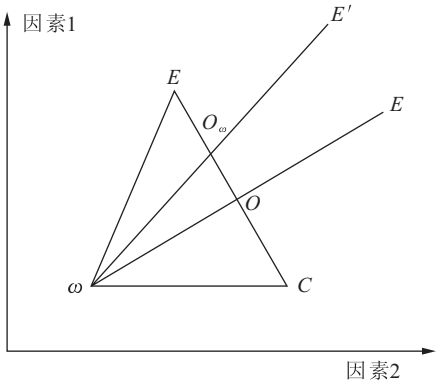


图 7－4　加权形心点

在上述三种单纯形算法的基础上，不少学者从不同角度对单纯形优化方法做了进一步研究，又相继提出了改进单纯形法（MSM）、超改进单纯形法（SMS）、控制加权形心超改进单纯形法（CWCSMS）和超改进控制加权形心单纯形法（SMCWC）等。Nelder－Mead 单纯形算法和加权形心单纯形法是目前被广泛采用的算法，其中 Nelder－Mead 算法又以其实现简单更受欢迎，这里采用的是 Nelder－Mead 算法。

7.1.4　算法的性能分析

1. 单纯形算法的优点

单纯形方法不同于沿着某一方向进行搜索的下山类优化方法，它向各个方向不断地优化新的单纯形，直至找到最优解。该算法对待估函数的连续性要求较低，不需要求其导数矩阵和 Hessian 矩阵，运算简单，比较适合求解大型和复杂函数表达式等计算量大的问题，所以在一些复杂的优化计算中，单纯形法比鲍威尔（Powell）算法得到更好的结果。

如果能够选择较合适的步长、延伸和压缩系数，单纯形法的迭代次数较少，收敛速度快，优于遗传算法等智能算法，比较适合单峰函数的优化求解。

2. 单纯形算法的不足

(1)初始点敏感。单纯形算法虽然简单高效,但它也并非适用于任何情况。对于一个给定的初始单纯形,单纯形算法经过迭代,能够快速地收敛到距离初始值最近的一个极值点,所以对于单调的函数而言,单纯形算法收敛效率很高,结果比较理想。但是,如果需要优化的函数不是单调的,而是对于给定的优化范围,存在多个极值点,则单纯形算法最终能收敛到哪个极值点取决于初始单纯形的选择。严格地讲,它会收敛到距离这个初始值较近的极值点,这就是单纯形算法所表现出来的初始点敏感问题,即:单纯形算法最终的优化结果将在很大程度上受到初始单纯形的选择。所以,从这个意义上讲,单纯形算法对于不严格单调的优化目标而言,缺乏全局优化能力,它只是一个局部搜索算法。

(2)振荡现象。单纯形算法在搜索的初始阶段通过反射、延伸等步骤能够快速地、以较大步长,以比较少的迭代次数接近最优解方向,但是在搜索的后期阶段,当试验点接近极小点时,其收敛速度明显变慢(耿长福,2006),甚至会出现围绕极小值点反复振荡而最终却收敛不到精确极小值点上的现象。

§7.2 单纯形算法的改进

7.2.1 振荡现象的实例分析

例如,对函数 $f_2 = 3x_1 \cdot x_1 + 2x_2 + 3x_2 \cdot x_2 - 4x_2$ 进行单纯形搜索。表 7-1 给出了单纯形算法在整个迭代过程中每一步迭代的单纯形顶点,以及这些点相对于目标点的偏差。从表中偏差的分布可以看出,在搜索的后期阶段偏差呈现正负的分布,揭示着其围绕最优解振荡的现象。

根据对大量的实例数据分析的结果可知,这种振荡的现象并非是偶然遇到的个别特例,而是一种由算法特性造成的,有一定的必然性。为了更清晰地表示这一特征,图 7-5、图 7-6 为当选择初始值为(0, 0)(−1, 1)时,使用 Matlab 绘制的算法的二维搜索路径,其中(a)图为搜索的全部过程,(b)图为(a)图中后期搜索产生振荡现象的部分放大,其中灰色“*”表示收敛的极小值点(0.000 489 5, 0.331 129)。从图中可以看出,当迭代进行一定步骤后,新确定的单纯形顶点多次跨越极小值点,呈现出围绕极小值点振荡的现象。

究其原因,延伸是在反射寻找到比现有顶点更优点的前提下,假定当前搜索方向为下降方向,为了最大化利用下降方向而进行继续优化的过程,如果延伸成功,则新形成的单纯形顶点无疑是一个较优点;然而压缩步骤是反射失败时所采取的补救措施, 如果反射失败,至少证明当前搜索方向不一定是最优下降方向,此时即使压缩,也容易错过寻找极小值的搜索方向, 而发生偏离。如果搜索过程中连续出现压缩, 则可能导致了极端情形下的振荡现象。

表 7－1　迭代数据列表

k	x_1	x_2	$f(x_1,x_2)$	x_1 偏差	x_2 偏差
1	1.000 000	－1.000 000	8.000 000	1.000 000	－1.333 333
1	1.500 000	－1.000 000	11.750 00	1.500 000	－1.333 333
1	1.000 000	－0.500 000	4.750 000	1.000 000	－0.833 333
2	0.000 000	－0.250 000	0.687 500	0.000 000	－0.583 333
2	0.000 000	0.250 000	－0.312 500	0.000 000	－0.833 333
3	0.300 000	0.150 000	0.037 500	－0.300 000	－0.183 333
4	－0.195 000	0.335 000	－0.219 250	－0.195 000	0.299 833
4	0.105 000	0.435 000	－0.269 250	0.105 000	0.101 667
5	－0.021 750	0.340 250	－0.331 771	－0.021 750	0.006 917
6	0.023 887	0.337 087	－0.331 579	0.023 887	0.003 754
7	0.000 748	0.312 068	－0.331 975	0.000 748	－0.021 265
8	－0.000 184	0.329 438	－0.333 288	－0.000 184	－0.003 895
9	－0.006 328	0.326 602	－0.333 077	－0.006 328	－0.006 731
10	－0.004 457	0.332 805	－0.333 273	－0.004 457	－0.000 528
11	0.001 686	0.335 641	－0.333 309	0.001 686	0.002 348
12	－0.000 812	0.332 619	－0.333 330	－0.000 812	－0.000 714
13	0.000 251	0.332 722	－0.333 332	－0.000 611	－0.000 611
14	0.000 309	0.333 562	－0.333 333	0.000 309	0.000 229
15	0.000 047	0.332 988	0.333 333	－0.000 047	－0.000 345

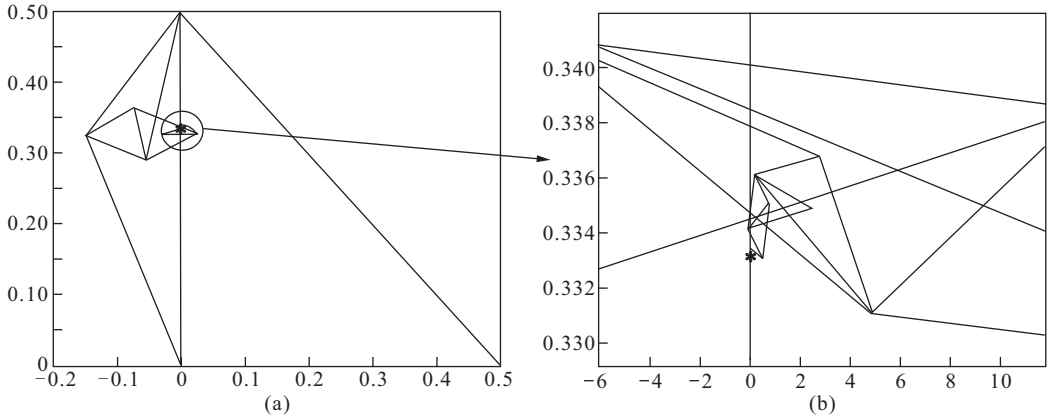


图 7－5　初始点(0，0)

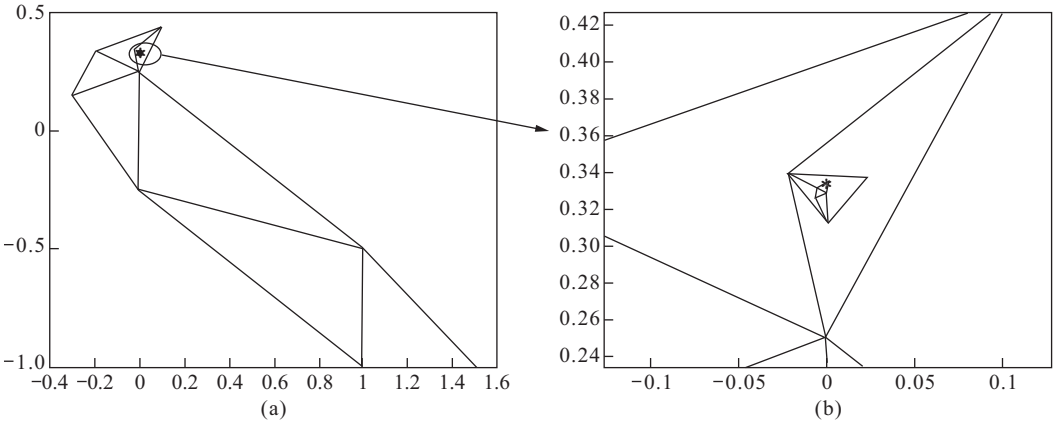


图 7-6 初始点(−1, 1)

7.2.2 算法改进策略

为了克服这种现象,这里对原始单纯形算法进行一定的修改,改变原压缩策略为:当反射点大于次大点时,不进行压缩,而是修改单纯形的形状直接令最大值点、次大值点以一定的比例靠近最小值点。算法步骤如下:

(1)构造初始单纯形。选取 $x^{(0)}$ 为起点,取一正数 h 为步长,令 $x^{(i)} = x^{(0)} + he^{(i)}$,其中 $e^{(i)}$ 为第 i 个单位坐标向量 $i = 1,2,\cdots,n$ 。

计算各顶点的函数值 $f(x)$:$f_i = f(x^{(i)})$,$i = 1,2,\cdots,n$。然后比较诸函数值的大小,选出最小值点 $x^{(L)}$、最大值点 $x^{(H)}$(对目标函数求极小而言)及次大值点 $x^{(G)}$,并计算其函数值 $f_L = f(x^{(L)})$、$f_H = f(x^{(H)})$、$f_G = f(x^{(G)})$。

(2) 求反射点 $x^{(R)}$。计算除去最大值点 $x^{(H)}$ 以外的 n 个点的形心 $x^{(C)}$,再进行反射,得到反射点 $x^{(R)} = x^{(C)} + \alpha(x^{(C)} - x^{(H)})$,其中 α 为反射系数,一般取 $\alpha = 1$。

反射后比较 f_R 与 f_L、f_G 有三种情形:如果 $f_R < f_L$,则转入步骤(3);如果 $f_R > f_G$,则转入步骤(5);否则转入步骤(4)。

(3) 延伸。若 $f_R < f_L$,则进行延伸,即令 $x^{(S)} = x^{(C)} + \beta(x^{(R)} - x^{(C)})$,其中 β 为延伸系数。计算延伸点函数值 f_E,若 $f_E < f_R$,则以点 $x^{(S)}$ 取代 $x^{(H)}$,构成新的单纯形,转步骤(6);否则以点 $x^{(R)}$ 取代 $x^{(H)}$,转步骤(6)。

(4) 直接替换。若 $f_L \leqslant f_R \leqslant f_G$,则以点 $x^{(R)}$ 取代 $x^{(H)}$,转步骤(6)。

(5) 缩进。若 $f_R > f_G$,则不进行压缩,令最差点 f_H 与次差点 f_G 分别向最小点 f_L 缩进:$x^{(i)} = x^{(i)} + \gamma(x^{(L)} - x^{(i)})$,$i = 1,2,\cdots,n$。其中 γ 为缩进系数,一般取 $0.6 \leqslant \gamma \leqslant 0.8$ 为宜。同时,缩小延伸系数 β 为原来的 0.9 倍,即 $\beta = \beta \times 0.9$。然后转步骤(6)。

(6) 检验。是否满足收敛准则:$\{\frac{1}{n+1}\sum_{i=1}^{n+1}[f(x_i) - f(\dot{x})]^2\} < \varepsilon$,若满足则停止计算,否

则转步骤(2)。

7.2.3　实验数据分析

为了验证算法的可行性，采取以下函数为算例对原算法与改进的算法进行比较：

$$f_1 = (x_1 - 3)^2 + 2(x_2 + 2)^2 ;$$

$$f_2 = 3x_1 \cdot x_1 + 2x_2 + 3x_2 \cdot x_2 - 4x_2 ;$$

$$f_3 = 4x_1 \cdot x_1 + 4x_2 \cdot x_2 - 4x_1 \cdot x_2 - 12x_2 ;$$

其中函数 f_1、f_2 和 f_3 的最优解分别是：0、−0.333 33 和−12(表 7-2)。

从表 7-2 的数据可以看出，在初始值距离最优解比较近时，改进的算法不仅迭代次数少，且收敛结果精确度比原算法高。

图 7-7、图 7-8 是函数 f_2 在初始点分别为(0,0)、(1,−1)时原算法与修改算法的二维搜索路径的对比[其中(a)为原算法，(b)为修改算法]，图中反映出当试验点已经接近极小值时，原算法围绕极值点波动很大；而改进的算法则能比较理想地克服这一情况，按照一定的规律循序渐进地靠近最优解，能够避免围绕极值点反复振荡，从而稳定、快速地收敛到极小点。

表 7-2　修改算法与原算法的对比

函数	初始值	函数值		迭代次数	
		原算法	修改算法	原算法	修改算法
f_1	(2.5, −2.5)	5.055 51e−005	5.792 92e−006	8	5
	(2, −1)	0	0	18	6
	(0, 0)	5.202 54e−005	1.354 26e−005	18	13
	(1, −1)	9.208 13e−005	1.818 51e−005	9	15
f_2	(0, 0)	−0.333 31	−0.333 33	10	5
	(1.5, 1)	−0.333 13	−0.333 33	16	12
	(3, 5)	−0.333 33	−0.333 32	26	18
f_3	(0, 1)	−12	−12	16	12
	(2, 3)	−11.998	−12	18	13
	(−1, 0)	−11.999	−12	18	12

注：其中收敛精度 $\varepsilon = 1e-4$，延伸系数为 1.7，原算法中压缩系数为 0.3。

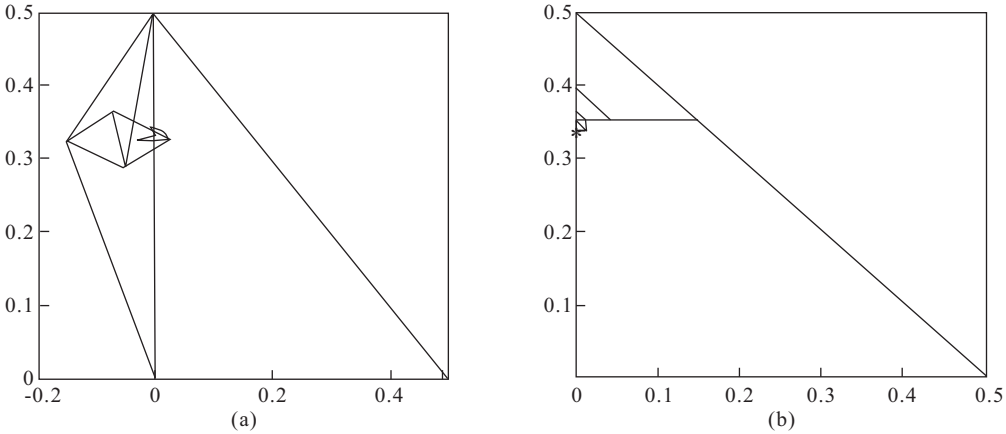


图 7－7　初始点为(0，0)的对比

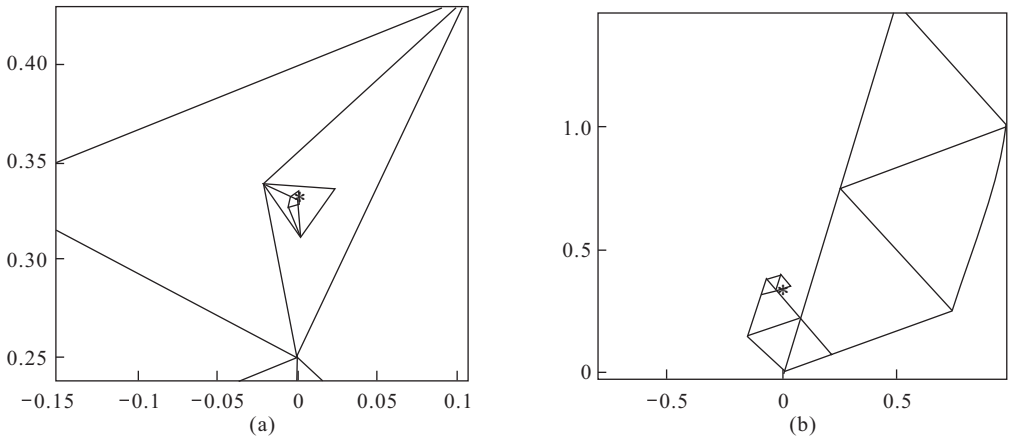


图 7－8　初始点为(1，－1)的对比

7.2.4　结果分析

书中针对试验点接近极小值点时原算法搜索效率降低的情况，不是通过取消原算法中的压缩策略，而是直接进行向最小值点适当靠近，并缩小扩张系数。经过大量的数值检验，证明这种方法在初始值距离最优解较近时能够取得优于原算法的效果。

但这种修改也具有一定的局限性，大量的实验数据证明，当初始值选择距最优解较远时，改进算法平均需要较多的步骤收敛，稳定性差于原算法，所以改进算法适宜在特定的前提下应用。

§7.3　遗传算法-单纯形两级优化算法

关于遗传算法与单纯形算法的结合策略，不少学者已经做过深入研究。在研究前人工作的基础上，认真分析了单纯形算法、遗传算法以及目标函数的特性，提出了一种对目标函数首先应用遗传算法进行全局优化，再应用改进单纯形算法进行局部优化的算法过程。

7.3.1　算法原理

下面分别对遗传算法与单纯形算法进行分析，构造一种新的两级优化算法。

1. 遗传算法

(1)遗传算法的全局搜索能力。遗传算法采用简单的编码技术表示各种复杂的结构，它采用种群的方式组织搜索，具有隐含的并行性，使得它可以同时搜索解空间的多个区域，像网一样覆盖整个搜索空间，从而有效防止了算法陷入局部最优。遗传算法的全局搜索能力是它优于传统优化算法的明显特性之一。

(2)遗传算法的效率。在另一方面，遗传算法对多个个体同时操作，而且遗传算法的随机转移规则也降低了算法收敛的速度，使得遗传算法没有传统确定性算法那么简单高效。

(3)遗传算法的精确度。由于遗传算法是随机搜索算法，大量的实验表明，在算法执行的后期算法收敛慢，甚至无法达到问题的精确最优解。

2. 单纯形算法

(1)单纯形算法简单、高效、精确的性能。非线性规划单纯形优化算法计算简单，收敛快速，精确度高，是一个性能优秀的局部搜索算法。

(2)单纯形算法的初始值敏感问题。也正因为这个算法是局部搜索算法，只适用于单峰函数的优化，对于存在多个极值点的目标函数，它会依据不同的初始值收敛到特定的某个极值解附近，所以对于多峰多谷函数的优化，一个好的初始点对于单纯形算法是至关重要的。

综上所述，结合遗传算法的全局搜索性能和单纯形算法局部优化性能，即利用遗传算法初步搜索到全局解附近，再用单纯形算法进行搜索，可以在降低计算量的同时保证结果的精确度。这样既克服单纯形算法初始点敏感，又克服了遗传算法效率不高的问题，能够有效地发挥两类算法的优点，形成一个高效的新算法。

7.3.2　算法实现

遗传算法-单纯形两级优化算法流程，如图 7－9 所示。

算法实现步骤如下：

(1)用遗传算法对目标函数进行全局优化。如果仅用遗传算法对目标函数进行优化，则一般进化代数较大，耗时较多，而此处设置较少的进化代数。

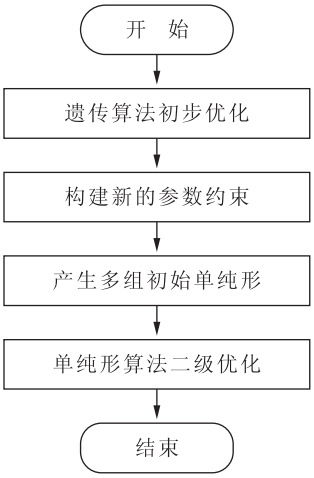


图 7-9 遗传算法-单纯形两级优化算法流程图

(2)以遗传算法的优化结果为中心构造一个与原参数约束成正比的新的参数约束作为单纯形算法的二级优化初始范围。

(3)对新参数约束应用改进的单纯形算法进行两级优化。为了避免初始点敏感问题,随机产生若干组初始单纯形,针对新的参数约束同时执行多次单纯形搜索,选取优化结果最好者作为最终结果。

7.3.3 实验分析

表 7-3、表 7-4 从不同角度给出了遗传算法分别与原单纯形算法、修改单纯形算法结合优化地球-火星转移轨道的数据结果。其中利用遗传算法初步优化的结果为极值点:(1 250.000,205.900),极值:5.681,迭代次数:750;单纯形算法与修改单纯形算法的延伸系数均为 1.7,单纯形算法压缩系数为 0.3,修改单纯形算法缩进系数为 0.8。

因为存在尾期振荡的困扰,即使通过若干次的搜索,单纯形算法在遗传算法基础上搜寻到精确的极值点比较困难。单纯形算法在随机产生 16 组初始单纯形才首次寻找到目标解,表 7-3 列出了这 16 次的寻找结果和取相同初始单纯形组数的修改单纯形算法的数据结果,其中产生随机初始值的参数约束为 $t_0 \in [1\ 200, 1\ 300]$ MJD2000,$\Delta t \in [195, 210]$Days。N 为 16 次单纯形算法,A 为每次搜索总迭代次数。

表 7-4 给出了遗传算法分别与原单纯形算法、修改单纯形算法结合,执行不同次数单纯形算法进行二级优化时,最终寻找到精确最优解的概率对比。

表 7－3　多组初始单纯形并行搜索结果原始单纯形与改进单纯形算法对比

N	单纯形算法优化结果（MJD2000，days，km/s）			A	修改单纯形算法优化结果（MJD2000，days，km/s）			A
1	1 256.199 944	195.445 280	5.703 383	16	1 253.436 260	203.666 078	5.669 622	20
2	1 252.443 116	206.994 012	5.675 385	22	1 253.425 348	203.698 133	5.669 623	25
3	1 252.412 109	208.000 000	5.679 402	17	1 253.420 654	203.669 126	5.669 622	21
4	1 254.208 986	203.736 732	5.670 280	26	1 253.454 885	203.631 371	5.669 623	26
5	1 253.458 257	203.659 74	5.668 620	22	1 253.433 169	203.664 778	5.668 620	27
6	1 254.081 128	201.298 398	5.672 428	26	1 253.440 484	203.679 940	5.668 620	24
7	1 254.979 393	211.011 767	5.708 465	22	1 253.437 850	203.663 104	5.668 620	25
8	1 256.058 076	196.455 751	5.696 048	18	1 253.129 400	205.778 200	5.671 945	10
9	1 253.952 864	201.724 239	5.671 511	28	1 253.441 482	203.663 551	5.668 620	28
10	1 265.675 000	214.270 000	5.923 619	6	1 253.432 849	203.658 634	5.668 620	20
11	1 233.275 027	222.659 755	6.025 236	19	1 253.424 334	203.645 716	5.669 623	15
12	1 254.294 765	200.759 043	5.673 862	18	1 253.435 977	203.665 026	5.668 620	21
13	1 253.722 917	201.513 348	5.671 990	20	1 256.096 000	196.363 200	5.696 733	7
14	1 255.035 878	196.971 002	5.691 608	21	1 253.424 844	203.686 516	5.668 620	21
15	1 254.100 000	196.050 000	5.699 534	16	1 253.441 471	203.666 262	5.668 620	17
16	1 228.000 000	200.000 000	6.391 042	2	1 253.438 385	203.673 162	5.668 620	27

表 7－4　不同初始组数两种算法搜索到目标解的对比

初始组数	单纯形算法		修改单纯形算法	
	搜索到目标解的次数	概率(%)	搜索到目标解的次数	概率(%)
2	0	0	2	100
4	0	0	3	75
6	0	0	6	100
8	1	12.5	6	100
16	1	6.3	13	81.2
24	2	8.3	19	79.2
32	3	9.4	23	71.9
64	3	4.7	48	75.0
100	4	4.0	73	73.0

从表 7－3、表 7－4 可以得出以下结论：

(1)遗传算法-单纯形两级优化算法在遗传算法进化到较少的代数后，利用单纯形算法进行二级优化能够精确寻找到全局最优解，效率较高。

(2)利用修改的单纯形算法与遗传算法结合比原单纯形算法得到的优化结果在分布上更聚集于目标解，搜索到全局最优解的概率更高，所以修改的单纯形算法与遗传算法结合的两级优化算法更加稳定，效率更高。

§7.4 遗传算法-单纯形混合优化算法

在上一节中讨论了怎样结合遗传算法与单纯形算法的特性来高效、精确地定位到目标问题的全局最优解。但在许多情况下，如果能够在此基础上寻找到包含全局解在内的目标函数的所有极值解，可以为实际应用提供更好的参考价值。

在多峰多谷类函数优化问题中，从传统的确定性优化算法到现代演化算法，如何设计一种高效的全局优化算法一直是广受关注的热点问题。在现在的研究中，大部分算法最终的目的都是试图寻找到所有极值解中最优的一个，即：全局最优解。但在局部优化解较多的情况下，如优化范围内吸引盆个数多，全局优化解附近存在多个伴生解，这时确定算法寻找到的解是全局最优解还是局部最优解，就牵涉到算法可靠度的衡量。目前主要是通过测试函数来判断，或者是考察它对一个已知解的目标函数进行优化的过程中所表现的性能，根据相关知识和经验来推测该算法的可靠性。这时，如果有这样一种优化算法，能在搜索机制上保证它不是仅仅把目标锁定在唯一的全局最优解上，而能搜索到包括局部解在内的全部极值解，这样就可以对比所有极值解，找出全局最优解，这样就保证了算法的稳定性。

另一方面，由于受到多方面因素的影响，特别是工程可靠性的要求，算法理论上求得的全局最优解在实际的应用中并不一定就好于其他的局部极值解。在实际应用中，某些局部极值点可能有更重要的价值。从这种意义上说，如果能够找出目标函数所有的极值解，不仅能增加算法使用者对解的选择可能，而且有利于决策者能够统揽大局，分析寻找到更有实际应用价值的优化结果。

7.4.1 算法原理

把单纯形算法作为交叉算子融入遗传算法中是单纯形算法与遗传算法的众多结合策略之一。遗传算法的交叉操作随机性较强、方向性差，无法保证交叉得到的子代优于父代(即：所谓的非效率计算)。而单纯形算法是一种直接搜索算法，对于 N 维问题采用 $N+1$ 个初始点为顶点组成单纯形进行计算，通过反射、延伸等简单操作改良单纯形较劣顶点逐步逼近相对优点的方法。它搜索的方向性强，步长大，速度快，寻找下降方向的效率高。它的这些特性与遗传算法交叉操作的盲目性可以互补，把单纯形算法加入遗传算法，可以增强遗传算法的

收敛效果。

遗传算法初始个体规模大，计算量大，收敛速度慢，普通的交叉操作难以保证收敛的精确度，加入单纯形算法，可以在提高算法整体效率的同时保证收敛精度。

非线性单纯形算法是最优化理论中一类典型的直接搜索方法，采用的是一种“下山”策略，即寻找一个“下降”的方向爬到一个局部最优点，如果遇到一个谷底，则算法结束，它不具备全局搜索能力。所以在一个多极值的目标函数优化中，它对于全局最优解没有概念。而遗传算法从种群中的诸多个体开始搜索，并加入变异算子，具有并行性和随机性，是一种全局优化算法，可以对算法整体的方向性进行控制，防止算法陷入局部最优，引导算法进行全局搜索。

单纯形算法实现简单，计算量小。对于给定的初始点，经过较少的迭代便可以相对精确地收敛到初始点所在吸引盆中的某个极值点，这些极值点有三种可能：全局最优解、局部极值点或者是这些解的伴生解。从前面论述来看，无论是哪种情况，这些极值点都有保留的价值。

7.4.2　算法实现

算法的步骤如下：

(1)遗传算法种群大小为 POPSIZE，初始化种群 population[POPSIZE]及遗传算法、单纯形算法各参数；设置指针 point 指向 POPSIZE；遗传算法迭代次数 generation，单纯形算法执行次数 kSM。

(2)对 population[i]执行轮盘赌选择、交叉、变异策略，其中 i=0,1,2,…point－1。

(3)随机产生正数 j∈[0,point－1]及一个个体取代 population[j]。

(4)以个体 population[j]为中心点生成初始单纯形，执行单纯形算法，得到最优解。

(5)取一适当正数 d，若 |x－population[i]|＞d，(其中 i＝point，point＋1，…POPSIZE)，point —，置 population[point]为 x。

(6)重复步骤(3)～(5)kSM 次。

(7)重复步骤(1)～(6)generation 次。

流程图如图 7－10 所示。

7.4.3　实验分析

图 7－11 的优化范围为 $t_0 \in [800, 3\ 800]$MJD2000，$\Delta t \in [100, 400]$Days，图 7－11 中(a)、(b)、(c)、(d)分别为混合算法进化 5、10、20、50 代后的结果。

图 7－12 的优化范围为 $t_0 \in [3\ 800, 7\ 000]$MJD2000，$\Delta t \in [100, 400]$Days，图 7－12 中(a)、(b)、(c)、(d)分别为结合算法进化 5、10、20、50 代后的结果。

从图 7－11、图 7－12 的优化结果可以看出，遗传算法-单纯形混合算法经过 10 代的进

化就能够寻找到全局最优解，50 代的进化就能比较稳定地寻找到包括全局最优解在内的全部极值点。

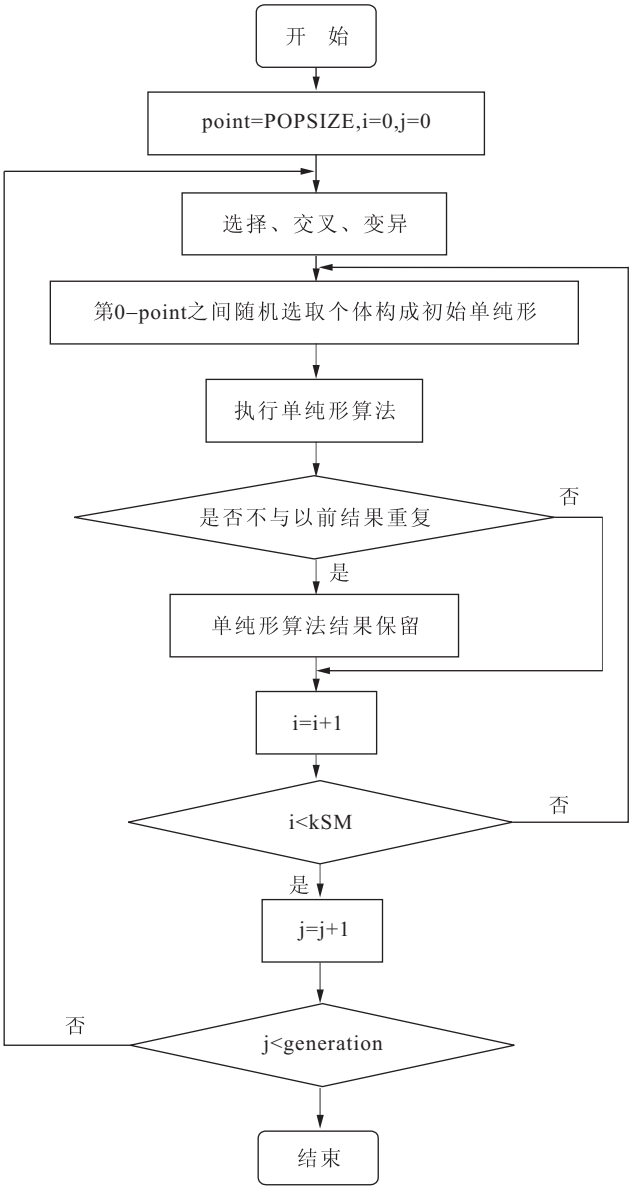


图 7－10 遗传算法-单纯形混合优化算法流程图

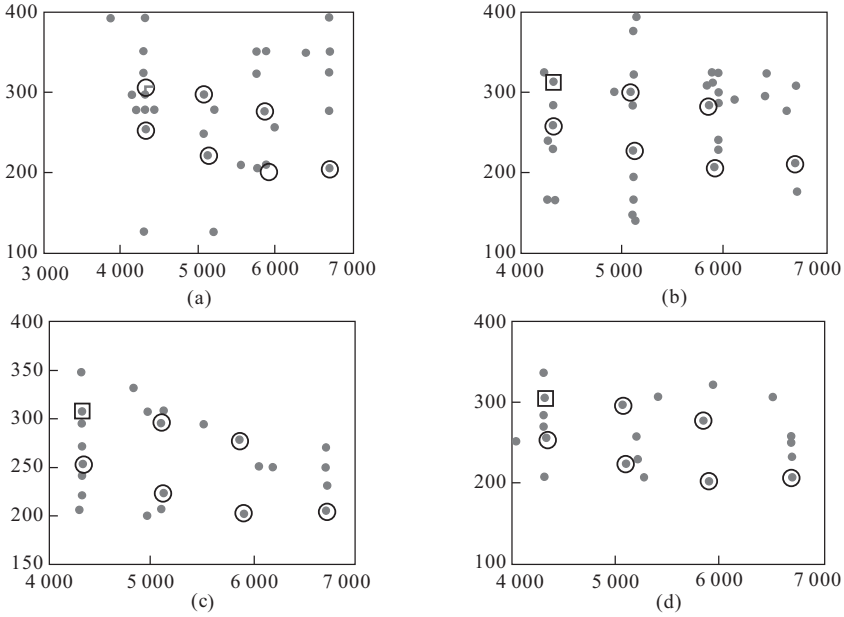


图 7－11　混合算法结果 $t_0 \in [800, 8\ 00]$MJD2000

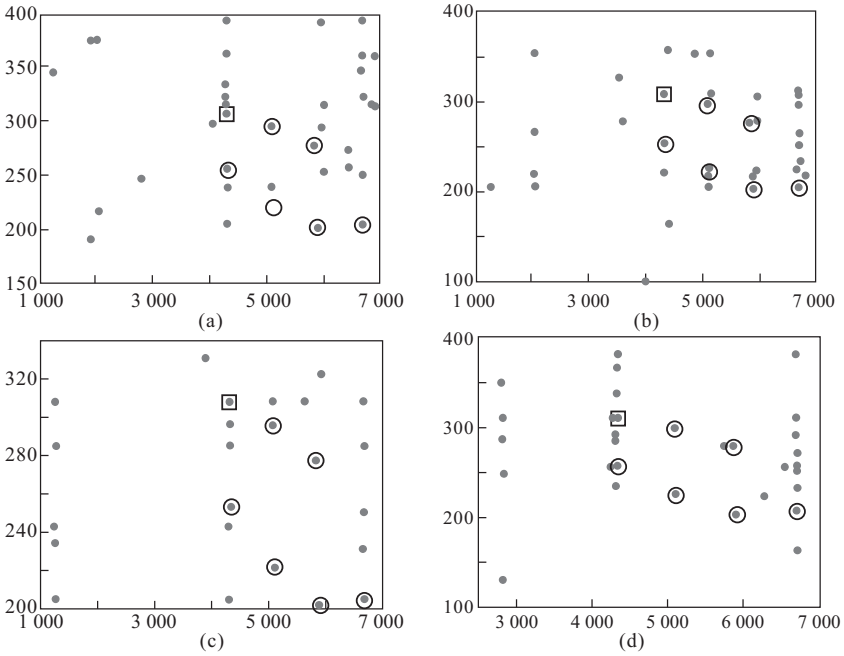


图 7－12　混合算法结果 $t_0 \in [3\ 800, 7\ 000]$MJD2000

表 7－5、表 7－6 给出了混合算法优化范围分别在 $t_0 \in [800,3\ 800]$MJD2000，$\Delta t \in [100,400]$Days 和 $t_0 \in [3\ 800,7\ 000]$MJD2000，$\Delta t \in [100,400]$Days 内，对表中所列的每一个进化代数(5、10、20、30、50、100)都执行 100 次，统计这 100 次中寻找到全部极值与全局最优解的次数而得到的概率。

表 7－5　优化性能列表 $t_0 \in [800,3\ 800]$MJD2000

优化范围(MJD2000，days)	进化代数	全部极值概率(%)	全局最优概率(%)
$t_0 \in [800,3\ 800]$ $\Delta t \in [100,400]$	5	18	93
	10	39	92
	20	68	98
	30	83	97
	50	79	100
	100	97	100

表 7－6　优化性能列表 $t_0 \in [3\ 800,7\ 000]$MJD2000

优化范围(MJD2000，days)	进化代数	全部极值概率(%)	全局最优概率(%)
$t_0 \in [3\ 800,7\ 000]$ $\Delta t \in [100,400]$	5	21	93
	10	51	92
	20	59	98
	30	76	97
	50	88	100
	100	94	100

图 7－13、图 7－14 的优化范围分别是 $t_0 \in [800,3\ 800]$MJD2000，$\Delta t \in [100,400]$Days 和 $t_0 \in [3\ 800,7\ 000]$MJD2000，$\Delta t \in [100,400]$ Days，图 7－13 中(a)、(b)分别表示搜索全部极值点和全局最优解的更精确的概率曲线图。图中每 5 代统计一次平均概率，从绘制的曲线走势可以看出，当进化代数大于 50 的情况下，算法寻找到全部极值点的概率大于 80%并趋于稳定，并且随着进化代数的增加，这个概率值将继续增大，并逐渐接近于 100%。而寻找到全局最优解的概率则更高。数据表明，当进化代数大于 20 时已可达到 100%，之后将一直稳定。

分析以上数据可得到以下结论：

(1)寻找全局最优解的稳定性：算法在运行 5 代后，寻找到全局最优解的概率已经达到 93%，之后随进化代数的增加，此概率保证稳定上升趋势，充分证明了其全局搜索能力较强。

(2)寻找全部极值解的稳定性：算法在运行 50 代后，寻找到全部极值点的概率超过 80%，作为随机搜索方法，已较稳定。

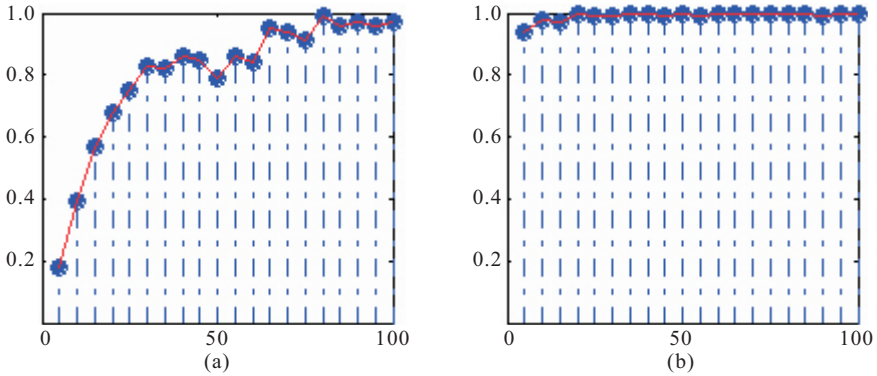


图 7－13　概率统计曲线 $t_0 \in [800,3\ 800]$MJD2000

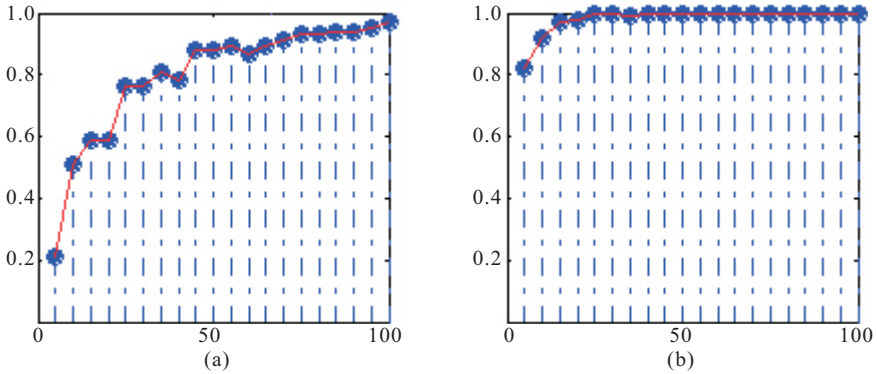


图 7－14　概率统计曲线 $t_0 \in [3\ 800,7\ 000]$MJD2000

遗传算法与单纯形算法优化原理不同，适用于求解不同特性的目标函数，因为自身特性的限制，都不能独立地求解目标问题。应该利用其互补性使其可以进行地球-火星双脉冲转移轨道能量的优化，并能得到满意的结果。

遗传算法-单纯形两级优化算法有效地结合了遗传算法的全局优化性能和单纯形算法的局部优化性能，在遗传算法全局优化的基础上运用单纯形算法进行局部优化，不仅减少了遗传算法的进化代数，而且提高了最终优化结果的精确度。

遗传算法-单纯形混合优化算法利用遗传算法全局控制能力，保证全局最优解能够被搜索到的同时，有效地融入单纯形算法，对单纯形算法的执行结果进行选择性地保存，这样不仅增加了搜索到全局最优解的可靠性，而且能够同时搜寻到解空间内的其他极值点。实验数据显示，种群大小相同时，这种混合策略仅需要 50 代的进化就能够搜索到包括全局最优解在内的全部极值解，充分证明了该算法的高效性。

第八章　双脉冲变轨问题的多目标优化算法

行星际转移轨道设计是行星探测任务设计的关键步骤之一，由于双脉冲控制设计简便，系统简单，易于实现，从地球到火星转移轨道通常采用双脉冲变轨。本章研究在设计地球到火星转移轨道时，同时考虑地球到火星轨道的转移时长即飞行时间 Δt 和两次脉冲所需能量 Δv_1 和 Δv_2 之和两个优化目标，即双脉冲变轨的多目标优化问题。

对此两个目标的优化问题，通常是把目标 Δt、Δv_1 和 Δv_2 进行聚合，把多目标优化问题转换成单目标优化问题，然后用序列二次规划 SQP 进行优化求解。其缺点是每次仅能得到一个可供选择的最优解。这里采用 NSGA－Ⅱ算法对飞行时间 Δt 和所需的总能量 Δv_1 和 Δv_2 两个优化目标同时进行优化，得到了一组可供选择的 Pareto 最优解集。

§8.1　地球到火星双脉冲转移轨道多目标优化模型

1. 从地球到火星双脉冲变轨设计需要优化的目标

(1) f_1＝飞行时间 Δt；

(2) f_2＝第一次脉冲所需的能量 Δv_1＋第二次脉冲所需能量 Δv_2 之和。

约束条件定义为：发射时刻 $t_0 \in [800, 3\ 800]$ MJD2000，飞行时间 $\Delta t \in [100, 400]$Days。

2. 优化问题的特殊性

第一：Δt 既是优化目标，同时又是目标 f_2 的优化参数。

第二：目标 f_2 是发射时刻 t_0 和飞行总时间 Δt 的函数，然而无法找到 f_2 的关于 t_0 和 Δt 的解析表达式。为求解 f_2 需要求解 Lambert 问题和 Kapler 方程。求解 Lambert 问题采用面积比迭代法。用穷举法画出的 f_2 的函数图 6－7，从图中可以看出该函数是一个多峰多谷的多极值函数。

§8.2　NSGA－Ⅱ算法

近年来，多目标进化算法在多目标领域已经成为一个研究热点，出现了许多优秀的算法，取得了较好的效果。其中非支配排序算法 NSGA－Ⅱ(Non－dominated Sorting Genetic Algorithm Ⅱ)是具有代表性的算法。NSGA－Ⅱ是 NSGA 算法的改进，在 NSGA 的基础上

加上了精英策略、虚拟适应度策略和快速非支配排序策略，在很大程度上改善了 NSGA 的缺点。

非支配排序遗传算法 NSGA 是由 Srinivas 和 Deb 于 1994 年提出的(Srinivas，*et al*，1994)。这是一种基于 Pareto 最优概念的遗传算法，它是众多的多目标遗传算法中体现 Goldberg 思想最直接的方法。该算法就是在基本遗传算法的基础上，对选择再生方法进行改进：将每个个体按照它们的支配与非支配关系进行分层，再做选择操作，从而使得该算法在多目标优化方面得到非常满意的结果。

虽然 NSGA 有一定的优点：优化目标个数任选，非劣最优解分布均匀，并允许存在多个不同的等价解。但它仍存在一些问题：①计算复杂度较高，算法复杂度为 $O(MN^3)$（其中 N 为种群大小，M 为目标函数的个数），当种群较大时，计算相当耗时；②没有精英策略，精英策略可以加速算法的执行速度，而且也能在一定程度上确保已经找到的满意解不被丢失；③需要指定共享半径 σ_{share}。

Deb 等(2000)针对 NSGA 的不足之处，提出 NSGA 的改进算法——带精英策略的非支配集排序遗传算法(NSGA－Ⅱ)。提出了非支配集排序的方法，以降低算法的计算复杂度。采用拥挤度距离，代替了需要指定共享半径的适应度共享策略，并在快速排序后的同级比较中作为胜出标准，使 Pareto 域中的个体能扩展到整个 Pareto 域，并均匀分布，保持了种群多样性。它采用了新的选择操作：在包含父种群和子种群的交配池中，依照适应度和分布度选择最好的 N(种群大小)个个体，从而使解有较好的分布性。它存在的不足是难以找到孤立点，此外当目标数增加时可能产生搜索偏移。NSGA－Ⅱ改进了原来算法的不足之处，提高了算法的运算速度和鲁棒性，并保证了非劣最优解的均匀分布。

8.2.1　快速的非劣解分类方法

为了根据个体的非劣解水平将种群分类，必须将每一个体与其他个体进行比较。NSGA－Ⅱ算法采用快速的非劣解分类方法，计算速度大大提高。

首先，对每一个解计算两个属性：① n_i，支配解 i 的解数目；②S_i，解 i 所支配解的集合。找到所有 $n_i = 0$ 的解并将其放入 F_1，称 F_1 是当前非劣解，其等级为 1。对当前非劣解中的每一个解 i，考察其支配集中 S_i 的每一点 j 并将 n_j 减少一个，如果某一个体 j 其 n_j 成为零，我们把它放入单独的类 H。如此反复考察所有的点，得到当前非劣解 H（级数为 2）。依次类推，直至所有解被分类。

8.2.2　虚拟适应度的计算

为了保持个体的多样性，防止个体在局部堆积，NSGA－Ⅱ算法首次提出了虚拟适应度(dummy fitness)的概念。它是指目标空间上的每一点与同等级相邻两点之间的局部拥挤距离。使用这一方法可以自动调整小生境，使计算结果在目标空间比较均匀地散布，具有较

好的鲁棒性。

个体之间的聚集距离如图 8－1，计算如下：

$$P[i]_{\text{distance}} = (P[i+1] \cdot f_1 - P[i-1] \cdot f_1) + (P[i+1] \cdot f_2 - P[i-1] \cdot f_2)$$

一般情况下，当有 r 个子目标时个体 i 的聚集距离为：

$$P[i]_{\text{distance}} = \sum_{k=1}^{r} (P[i+1] \cdot f_k - P[i-1] \cdot f_k)$$

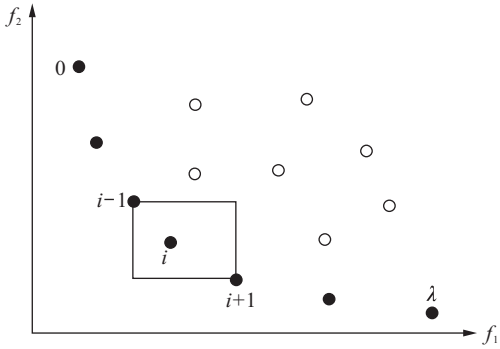


图 8－1　个体之间的聚集距离

8.2.3　选择运算

选择过程使优化朝 Pareto 最优解的方向进行并使解均匀散布。设每一个体的非劣等级属性和局部拥挤距离属性分别为 i_{rank} 和 i_{distance}，采用轮赛制，即随机选取两个个体并进行比较，保留一个优良个体，淘汰另一较差个体。具体过程为：

if($i_{\text{rank}} < j_{\text{rank}}$) or (($i_{\text{rank}} = j_{\text{rank}}$) and ($i_{\text{distance}} > j_{\text{distance}}$)) then $i > j$

or　else　$j > i$

比较两个个体，如果非劣等级不同，则取等级高（级数值小）的点。否则，如果两点在同一等级上，则取比较稀疏区域内的点，以使进化朝非劣解和均匀散布的方向进行。

8.2.4　精英保留策略

首先，将父体和子代全部个体合并成一个统一的种群 $R_t = P_t \cup Q_t$ 放入进化池中，种群 R_t 的个体数成为 $2N$。然后种群 R_t 按非劣解等级分类并计算每一个体的局部拥挤距离。依据等级的高低逐一选取个体直到个体总数达到 N，从而形成新一轮进化的父代种群 P_{t+1}，其个体数为 N。在此基础上开始新一轮的选择、交叉和变异，形成新的子代种群 Q_{t+1} 。这种方法可加快进化的速度，如图 8－2。

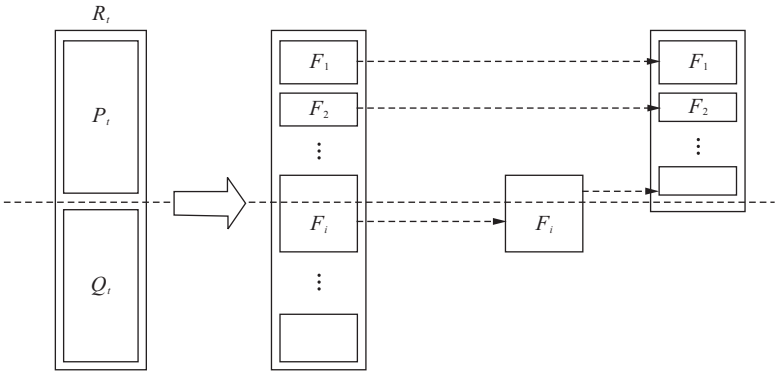


图 8-2　在组合后的种群中选择 N 个个体

8.2.5　遗传操作

1. 选择

在选择之前对种群进行非劣分层排序，每次随机地选择两个个体 i 和个体 j，如果个体 i 的 rank 少于个体 j 的 rank，则选择个体 i；否则选择个体 j。

2. 交叉

本书采用的是一种模拟二进制交叉(SBX)，从当代种群和交配池中分别选择其第 i 个体作为父体 1、父体 2，把父体 1 和父体 2 交叉后得到子体 1 和子体 2 的第 i 个决策变量的值按如下公式计算：

$$C_1^i = 0.5[(y_1^i + y_2^i) - \beta_{qi}(y_1^i - y_2^i)];$$

$$C_2^i = 0.5[(y_1^i + y_2^i) + \beta_{qi}(y_1^i - y_2^i)]$$

式中：C_1^i、C_2^i 分别为个体 1 和个体 2 的第 i 个决策变量的值；

y_1^i、y_2^i 分别为父体 1 和父体 2 的第 i 个决策变量的值；

$y_{\min}^i$、$y_{\max}^i$ 分别为父体 1 和父体 2 的第 i 个决策变量的下约束。

3. 变异

采用多项式变异：

$$\alpha = \begin{cases} (2\mu)^{\frac{1}{\eta+1}} - 1, & \mu < 0.5 \\ 1 - [2(1-\mu)]^{\frac{1}{\eta+1}}, & \mu \geqslant 0.5 \end{cases}$$

μ 是[0,1]之间的一个随机数，η 任意指定。个体变异后第 i 个决策变量的值为：

$$C'_i = C_i + \alpha(y_{\max} - y_{\min})$$

8.2.6 算法流程

随机产生一个规模为 N 的初始种群 P_0，将种群中的所有个体快速非支配排序。采用选择、交叉遗传算子产生一个规模为 N 的子代种群 Q_0。其中，选择算子主要根据累积排序值评价个体的优劣，选择累积排序值小的个体参与繁殖。将 P_0 和 Q_0 合并为一个规模为 $2N$ 的种群 R_0，对 R_0 进行非支配排序得到非支配个体集（F_1，F_2），选择前 i 个非支配集和 F_{i+1} 的前 $N-\sum_{j=1}^{i}|F_i|$ 个个体组成 P_1。$\sum_{j=1}^{i}|F_i|\leqslant N$，且 $\sum_{j=1}^{i+1}|F_i|>N$，再由 P_1 经选择、交叉产生 Q_1，将 P_1 和 Q_1 合并为 R_1。重复上面的循环，直到满足停止条件。算法流程如图 8－3。

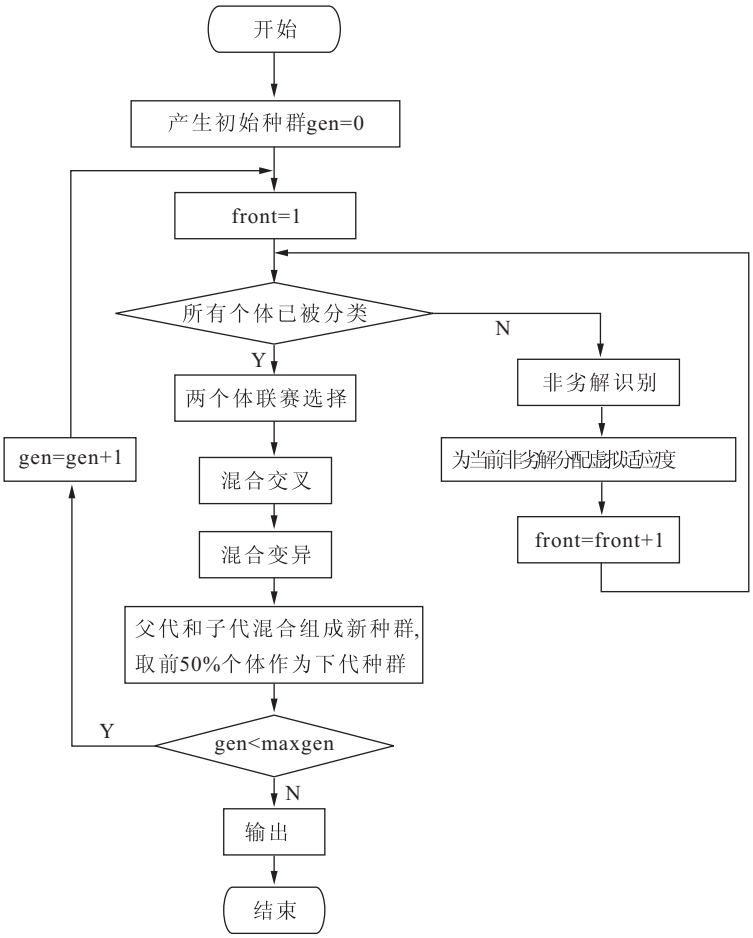


图 8－3 INSGA－Ⅱ流程图

§8.3　数值实验

8.3.1　参数设置

算法的参数设置：

种群大小	演化代数	交叉概率	变异概率	SBX－η_c	SBM－η_m
100	5 000	0.8	0.1	20	10

8.3.2　数值实验结果及分析

用 NSGA－Ⅱ算法，从演化后非劣解集中的 100 个体中选取以 $f_2(\Delta v_1+\Delta v_2)$ 为目标的最小 10 个解，如表 8－1。

表 8－1　NSGA－Ⅱ算法实验结果

最小点	t_0(MJD2000)	Δt(days)	$\Delta v_1+\Delta v_2$(km/s)
1	1 253.323	203.732 5	5.668 62
2	1 254.140 7	200.218 6	5.674 809
3	1 253.838 501	198.134 9	5.684 887
4	1 256.919 556	195.802 4	5.702 849
5	1 253.838 501	194.678 1	5.711 872
6	1 254.140 747	193.866 7	5.718 87
7	1 257.139 404	191.689 2	5.739 797
8	1 253.095 825	191.270 4	5.756 661
9	1 257.149 536	189.248 5	5.769 66
10	1 259.452 148	184.705 9	5.844 383

Myatt 等(2003)把优化目标飞行时间 Δt 和两次脉冲所需的能量 $\Delta v_1+\Delta v_2$ 聚合得到：

$$f=k\Delta t+l(\Delta v_1+\Delta v_2)$$

为了求得 f 局部最优解，将搜索空间每一维划分为 50 块，在每一个格网内，执行 SQP 算法，找到了对应网格的局部最小值，使用这种方法，找到 11 个不同的最小值，如表 8－2。

图 8－4 为所求得的 Pareto 前沿，图 8－5 为所求得发射时刻 t_0、飞行时间 Δt 以及两次脉冲所需的能量 $\Delta v_1+\Delta v_2$ 的三维图形。

从表 8－1 和表 8－2 可以看出，用 NSGA－Ⅱ算法优化从地球到火星双脉冲转移轨道能够近似找到 Myatt 等(2003)所提供的所需能量最小值。表 8－2 中的第 2～9 个解用 NSGA－Ⅱ算法是不可能得到的，因为它们被第 1 个解所支配。

但是如果把参数 t_0 的搜索空间分为三段，分别为[800，1 800]，[1 800，2 800]，[2 800，3 800]，取这三段空间的以 f_2 为目标的 5 个典型代表解，如表 8－3。

表 8－2　SQP 算法实验结果

最小点	t_0(MJD2000)	Δt(Days)	$\Delta v_1+\Delta v_2$(km/s)
1	1 253.7	202.9	5.667
2	3 573.4	323.2	5.673
3	2 812.5	344.2	6.223
4	1 225.5	233.2	6.353
5	2 057.5	216.5	6.835
6	3 597.8	273.5	6.861
7	2 048.2	350.5	7.106
8	2 834.3	246.0	7.369
9	3 766.8	400.0	21.633
10	800.0	400.0	44.292
11	3 800.0	126.7	63.289

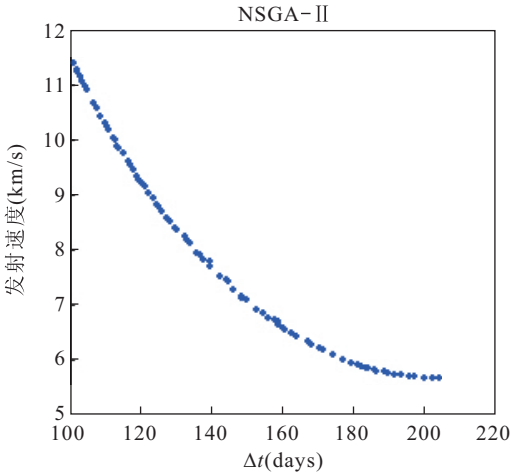


图 8－4　Pareto 前沿二维

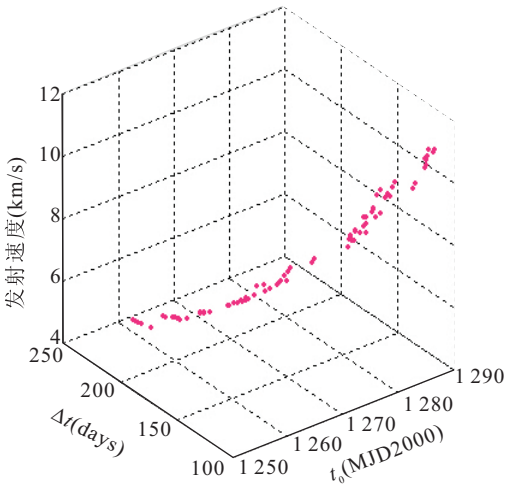


图 8－5　Pareto 前沿三维

表 8－3　把 t_0 的搜索空间分为三段，用 NSGA－Ⅱ算法求得的结果

区间	Minima	t_0(MJD2000)	Δt(days)	$\Delta v_1+\Delta v_2$(km/s)
1	1	1253.3239	203.7319	5.6686
	2	1255.0615	201.4479	5.6728
	3	1256.5621	195.3304	5.7049
	4	1257.1024	191.6095	5.7405
	5	1259.4315	182.2643	5.8991
2	1	2056.5358	215.2409	6.7881
	2	2057.0205	190.1514	7.1771
	3	2061.0171	182.2305	7.4577
	4	2058.9829	179.6598	7.5446
	5	2062.4592	174.1694	7.7928
3	1	3573.1176	324.1343	5.6694
	2	3574.3386	323.1188	5.6716
	3	3565.5385	290.5139	6.7126
	4	3598.2514	276.2226	6.8901
	5	2832.2187	235.6662	7.4478

比较表 8－2、表 8－3 可以看出，用 NSGA－Ⅱ算法把 t_0 的搜索空间分为三段，同样可以找到与文献(Myatt，*et al*，2003)很接近的解。

从地球到火星双脉冲转移轨道设计优化的两个目标是飞行时间 Δt 和两次脉冲所需的能量 $\Delta v_1+\Delta v_2$，这是一个典型的多目标问题。如按照马文臻(2006)把这两个目标聚合成单目标，通常是更多地考虑在两次脉冲所需的能量 $\Delta v_1+\Delta v_2$ 最省，而忽略了飞行时间 Δt 这个目标；而且每次仅可以得到一个可供选择的解。但是采用 NSGA－Ⅱ算法除了得到Myatt等(2003)所得的最优解外，还可以得到一组可供决策者参考的 Pareto 解集。

第九章　轨道设计与优化仿真工具

计算机仿真是通过建立数学模型，以程序的方式来模拟真实的事物，对目标进行快速准确分析的同时，具有节约投资、直观生动的优点，已经广泛应用于各个领域。计算机仿真技术也在航天领域得到广泛的应用，并为航天事业的宣传和技术应用等领域发挥着不可替代的作用。因此，在开展航天任务轨道设计和优化的研究工作中，引入可视化仿真技术也极具意义。

本章就设计和实现行星际轨道优化和仿真的工具—— Interplanetary Trajectroy Optimization Tool(ITOT)，从总体框架、数值计算和可视化仿真等方面展开介绍。

§9.1　总体结构

ITOT 的设计采用面向对象思想，采用 C++语言开发，同时按功能划分不同，可看作由不同的模块组成。其中核心类有 6 个，按模块划分，可将系统划分为 15 个子模块。

9.1.1　模块组织

整个优化仿真工具 ITOT，从功能模块上划分，可大致划分为 5 个层次和 15 个子模块，ITOT 的总体框架如图 9－1 所示。

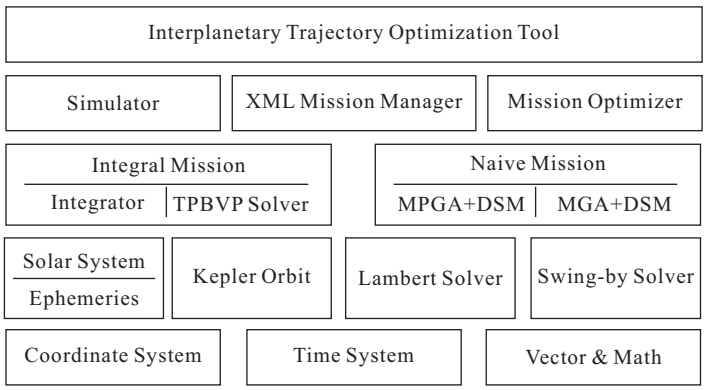


图 9－1　ITOT 总体框架

Coordinate System：坐标系统，实现日心惯性坐标系和行星中心惯性坐标系的转换；

Time System：时间系统，实现了时间不同儒略历元之间的转换，儒略日和 UTC 以及 TAI 之间的转换；

Vector & Math：矢量运算和数学库，提供所有矢量运算，包括矢量旋转等功能（ITOT 中未显示的使用矩阵方式来完成坐标旋转等功能）；

Solar System/Ephemeries：太阳系和星历模块，该模块提供太阳系模型中的所有常量和星历数据，同时实现了基于 ESA 和 JPL 的两种拟平均根数星历模型；

Kepler Orbit：基本二体轨道求解模块，维护一条二体轨道的根数等信息；

Lambert Solver：Lambert 问题求解模块，使用一个 TransOrbit 类来维护一条交会轨道的参数信息；

Swing - by Solver：借力飞行求解模块，求解带机动和无机动两种不同的借力飞行轨道模型；

Intergal Mission/Integrator/TPBVP Solver：数值积分轨道任务管理/多体问题轨道积分/两点边值问题求解。Integrator 根据给定的航天器位置速度初值，在日心惯性坐标系中进行轨道积分，实现了单步法如 RK4 和多步法如 Adams - Cowell。积分轨道交会问题（两点边值问题 Tow Point Boundary Value Problem，TPBVP）的求解，则通过打靶法和微分校正方法实现。

Naive Mission/MPGA＋DSM/MGA＋DSM：简单任务管理器，管理 MPGA＋DSM 和 MGA＋DSM 的行星际探测任务参数，并为上层优化和仿真提供访问接口；

Simulator：任务仿真模块，主要功能包括，使用 OpenGL 渲染背景，仿真太阳系行星运行过程，结合 XML Mission Manager，绘制一个行星际飞行任务的轨迹；

XML Mission Manager：任务想定管理器，所有的想定数据存放于 XML 文件中，该模块负责解析想定文件，并以树状结构向用户展示，提供编辑保持等功能；

Optimizer：任务优化模块，该模块对 MPGA＋DSM 和 MGA＋DSM 等类型任务的优化模型进行优化，以独立线程进行优化，并提供优化算法参数设置和优化进程监视等功能界面。

9.1.2　核心类简述

ITOT 的核心虚基类都以 AbstractXxx 的方式命名，以纯虚或虚函数方式向上层提供统一操作的接口，以增强软件可扩展性和开发的协同性。正如面向对象程序设计所提倡的，上层通过虚基类的指针透明地调用实例化的子类，而无需关心底层实现细节。下面就几个核心虚基类作简单介绍。

1. AbstractEphemerise

星历表类，管理不同的星历模型，以统一的接口方式向上层调用者提供不同的星历接

口，目前实现了 JPL 和 ESA 提供的两种星历模型（详见 § 9.2 节）。

2. AbstractMission

行星际探测任务类，分别实例化了简单任务类（基于二体）NaiveMission 和积分任务类（基于多体积分）IntegralMission。为飞行过程仿真模块（simulator）提供接口 getCraftPos-Vel，返回航天器在指定时刻的位置和速度矢量；为任务优化模块提供任务燃料消耗（总 Δv）计算接口 MissionEnergy，作为优化算法的任务评价函数。

3. AbstractOptimizer

虚优化器类，已实现 DeOptimizer 和 PsoOptimizer 分别是差分演化（DE）和粒子群算法（PSO）优化器，任务优化模块（Optimizer）提供迭代一步（IterateOneStep）的接口，以监视优化进程，同时还提供其他的优化控制接口。

4. AbstractTrajectory

已实例化子类开普勒轨道类（KepOrbit）和积分轨道类（IntegralTrajectory）分别维护开普勒轨道和积分轨道信息，实现向外部提供任意时刻航天器位置矢量和速度矢量的接口 getPosVel，以统一的操作方式为可视化仿真模块（Simulator）提供轨迹信息。

5. AbstractIntegrator

积分器类，已实例化子类 Rk4Integrator 和 Ac4Integrator，分别为四阶 Runge - Kutta 和四阶 Adams - Cowell 方法积分器，Integrate 为上层提供积分操作接口。

6. AbstractTPBVSolver

积分轨道交会问题（两点边值问题）求解器，本书尝试了基于状态转移矩阵的微分校正方法和基于遗传算法的打靶法，试图从精确数值计算上对行星际探测任务的轨迹进行仿真。

§ 9.2 数值仿真

对行星际轨道任务的数值仿真，便于与实际任务或其他文献中提供的任务数据进行比较，并方便将轨道设计和优化的结果进行分析或改进。本系统的数值仿真主要涉及星历计算，基于二体和多体积分的航天器轨迹仿真等。

在航天器轨道理论中，除二体问题等少数几种情况外，都不可能给出严格的解析解，所以在精度要求较高时通常采用数值算法。轨道交会问题，在二体模型中可以使用 Lambert 方法进行求解；而在多体模型中，则必须通过打靶或微分校正等方法进行迭代求解。

9.2.1 星历计算

太阳系行星的运行规律非常复杂，高精度的星历计算通常是一个复杂的查表和插值的过程，在深空探测精确导航中使用。自 20 世纪 70 年代初，来自 JPL 和 MIT 的星历已经成为世界标准，目前较流行的星历模型是 JPL 的 DE 系列。

此外，JPL 和 ESA 都提供了一些近似星历模型（Standish，2008；ACT，2008），这些模型采用拟平均根数的方法实现星历的近似计算，其中 ESA 模型是基于 JPL－DE405 星历模型的五次多项式拟合所得。这些虽然是近似的星历模型，但在初轨优化设计中，尤其是圆锥曲线拼接法的模型中，其精度已经足够。因此本系统使用的是 JPL 和 ESA 的星历模型。

考虑到软件可扩展性能，本书的星历计算模块独立封装，可以在软件运行过程中随时更换星历模型，也可以针对不同的任务应用不同的星历模型。星历模型提供如式（9－1）的星历计算接口。

$$(r_{\mathrm{Pl}}, \dot{r}_{\mathrm{Pl}}) = \mathrm{eph}(\mathrm{Pl}, t) \tag{9-1}$$

式中：Pl 为行星 ID；

t 用于指定时刻。

9.2.2　二体任务仿真

简单任务 Naive Mission 基于二体轨道模型，分为 MPGA＋DSM 和 MGA＋DSM 两种类型。Naive Mission 使用一个 Manoeuvre 序列来维护多次轨道机动信息，不同的机动类型描述如表 9－1 所示。

表 9－1　NaiveMission 轨道机动类型

Manoeuvre 类型	参　数	说　明
Departure	Pl，t，v	发射（仅考虑逃逸双曲线剩余速度）
Dsm	DsmPos，t，v	深空机动
Powered Swing－by(PGA)	Pl，t，v	带动力借力飞行
Swing－by(GA)	Pl，t，r_p，θ，v	纯借力飞行
Landing	Pl，v	着陆（仅考虑飞入双曲线剩余速度）
Off Orbiting	Pl，t，r_p，e，v	脱离绕飞状态
To Orbiting	Pl，t，r_p，e	进入绕飞状态

表中的参数：

Pl 为太阳系中的行星 ID；t 为表示脉冲机动发生的时刻；DsmPos 为深空脉冲机动在日心惯性坐标系 $O_s x_s y_s z_s$ 中的位置（对于 MGA＋速度模型 DSM 的 NaiveMission，需要根据式（2－26）解出 DSM 位置）；r_p 为指借力飞行或绕飞轨道的近心点高度（km）；θ 为无机动借力飞行过程中的 B 平面旋转角；e 为脱离前或进入后的行星中心绕飞轨道偏心率；v 为机动后航天器的日心轨道速度，在 MGA＋速度模型 DSM 的 Departure 机动类型中，v 为指定量，见式（4－8）；其他机动类型的 v 则需要求解 Lambert 问题得出，见式（3－14）。

若已知一个行星际探测任务的机动序列，则根据二体轨道方程和对 Lambert 问题的求解，不难得出任意时刻航天器在日心惯性系中的位置和速度；同时也不难计算出各次机动的 Δv 大小，作为优化算法的目标函数之一。

9.2.3 多体任务数值仿真

在§2.2 节中，介绍了多体问题的运动方程。从式(2－37)不难发现，在多体模型中，航天器的加速度 $\ddot{\vec{r}}$ 是关于航天器所在日心惯性系中与其他行星相对位置的函数。结合已有的星历模型，见式(9－1)，将航天器运动方程变形为如下形式：

$$\ddot{\vec{r}} = f(\vec{r}, t) \tag{9-2}$$

若已知航天器在 t_0 时刻在日心惯性坐标系中的位置矢量 $\vec{r}$ 和速度矢量 $\dot{\vec{r}}$，则航天器运行轨迹的计算实际上就是一个常微分方程的初值问题。求解多体积分轨道的首要任务是积分算法的选择，包括单步法和多步法。

单步法以 Runge－Kutta(RK)方法最为常用，它使用方便，且积分起步容易，同时又可以变步长计算，其缺点是当方法阶数增高，其右函数的计算量也相应增大。高阶的常微分方程初值问题通常可转换为一阶方程组的方式进行求解，二阶的航天器运动方程可作如下转换(沈剑华，2000)：

$$\begin{cases} \dfrac{\mathrm{d}\dot{\vec{r}}}{\mathrm{d}t} = f(\vec{r}, t) \\ \dfrac{\mathrm{d}\vec{r}}{\mathrm{d}t} = \dot{\vec{r}} \end{cases} \tag{9-3}$$

则四阶 RK 具有如下形式：

$$\begin{cases} \dot{\vec{r}}_{n+1} = \dot{\vec{r}}_n + \dfrac{h}{6}(\vec{k}_1 + 2\vec{k}_2 + 2\vec{k}_3 + \vec{k}_4) \\ \vec{r}_{n+1} = \vec{r}_n + h\dot{\vec{r}}_n + \dfrac{h^2}{6}(\vec{k}_1 + \vec{k}_2 + \vec{k}_3 + \vec{k}_4) \end{cases} \tag{9-4}$$

其中：

$$\begin{cases} \vec{k}_1 = f(\vec{r}_n, t_n) \\ \vec{k}_2 = f\left(\vec{r}_n + \dfrac{h}{2}\dot{\vec{r}}_n, t_n + \dfrac{h}{2}\right) \\ \vec{k}_3 = f\left(\vec{r}_n + \dfrac{h}{2}\dot{\vec{r}}_n + \dfrac{h^2}{4}\vec{k}_1, t_n + \dfrac{h}{2}\right) \\ \vec{k}_4 = f\left(\vec{r}_n + h\dot{\vec{r}}_n + \dfrac{h^2}{2}\vec{k}_2, t_n + h\right) \end{cases} \tag{9-5}$$

多步法常用 Adams－Cowell(AC)方法，它将 Adams 和 Cowell 方法联合使用(付兆萍

等，2006)，即 Adams 方法提供速度矢量 $\dot{\vec{r}}$，同时由 Cowell 方法计算位置矢量 $\vec{r}$。在具体计算时，常采用 Adams 预报-校正算法，即显式和隐式相结合的方法。本书中使用的是四步四阶显式公式作为预报公式，三步四阶隐式公式作为校正公式，其他阶的参数以及推导方法见文献(刘林，2000；刘强等，1998)。

则 AC 预报公式为：

预测：
$$\dot{\vec{r}}_{n+1}^{p} = \dot{\vec{r}}_{n} + \frac{h}{24}(55f_{n} - 59f_{n-1} + 37f_{n-2} - 9f_{n-3})$$

修正：
$$\dot{\vec{r}}_{n+1}^{pm} = \dot{\vec{r}}_{n+1}^{p} + \frac{251}{270}(\dot{\vec{r}}_{n}^{c} - \dot{\vec{r}}_{n}^{p}) \tag{9-6}$$

$$\vec{r}_{n+1} = 2\vec{r}_{n} - \vec{r}_{n-1} + \frac{h^{2}}{60}(68f_{n} - 19f_{n-1} + 14f_{n-2} - 3f_{n-3})$$

校正公式：

校正：
$$\dot{r}_{n+1}^{c} = \dot{r}_{n} + \frac{h}{24}[9f(r_{n+1}^{pm}, t_{0} + (n+1)h) + 19f_{n} - 5f_{n-1} + 2f_{n-2}]$$

修正：
$$\dot{r}_{n+1} = \dot{r}_{n+1}^{c} - \frac{19}{270}(\dot{r}_{n+1}^{c} - \dot{r}_{n+1}^{p}) \tag{9-7}$$

$$r_{n+1} = 2r_{n} - r_{n-1} + \frac{h^{2}}{60}[3f(r_{n+1}, t_{0} + (n+1)h) + 56f_{n} - f_{n-1} + 2f_{n-2}]$$

式中：t_0 为积分初始时刻；

h 为积分步长。

§9.3　任务管理与可视化

本系统的可视化仿真部分基于 OpenGL 实现三维场景渲染，实现任务轨迹仿真和星空背景仿真、太阳系主要天体仿真等。

9.3.1　任务管理器

XML Mission Manager 采用 XML 文件方式维护任务信息，采用 DOM(Document Object Model)技术对 XML 文件进行操作，虽然基于 DOM 的数据模型具有对超大文档的处理效率低下的缺点，但考虑本书的数据量较少，仍采用这一便于理解和处理的面向对象模型。

存储任务想定数据的 XML 文件的根为“<xmission>”节点，每个“<misson>”节点包含三个属性，子节点为“<manoeuvre>”用于维护表 9-1 中的不同机动类型。“<manoeuvre>” 节点包含三个可选属性，或一个子节点，子节点可以是“<position>”用于指定 DSM 的位置，也可以是“<orbiting>”用于指定绕飞轨道的相关参数，如绕飞行星，绕飞轨道高度

和偏心率等。

下面给出的是一个完整的 XML 文档内容，该文档包含一个任务（“〈mission〉”），任务名称为“Cassini”，该任务共 6 次机动，其中包括 4 次带机动的借力飞行（PGA）和 1 个逃逸机动以及 1 个进入土星绕飞轨道的机动。

```
<?xml version="1.0" encoding="iso-8859-1"? >
<xmission>
  <mission caption="Cassini" category="naive" ephemeris="esa" >
    <manoeuvre category="Departure" planet="Earth" mjd2k="-789.811722" />
    <manoeuvre category="Powered Swingby" planet="Venus" mjd2k="-631.509695" />
    <manoeuvre category="Powered Swingby" planet="Venus" mjd2k="-182.123821" />
    <manoeuvre category="Powered Swingby" planet="Earth" mjd2k="-127.374852" />
    <manoeuvre category="Powered Swingby" planet="Jupiter" mjd2k="896.987206" />
    <manoeuvre category="To Orbiting" mjd2k="5449.295174" planet="Saturn" >
      <orbiting eccentricity="0.98" pericenter_height="108950" />
    </manoeuvre>
  </mission>
</xmission>
```

在软件界面上，以一个树状结构呈现整个任务的参数等信息（包括使用哪种星历模型、机动次数、每次机动的时刻等），并提供任务编辑保存等功能和界面，前述“Cassini”任务的数据管理器界面见图 9－2。

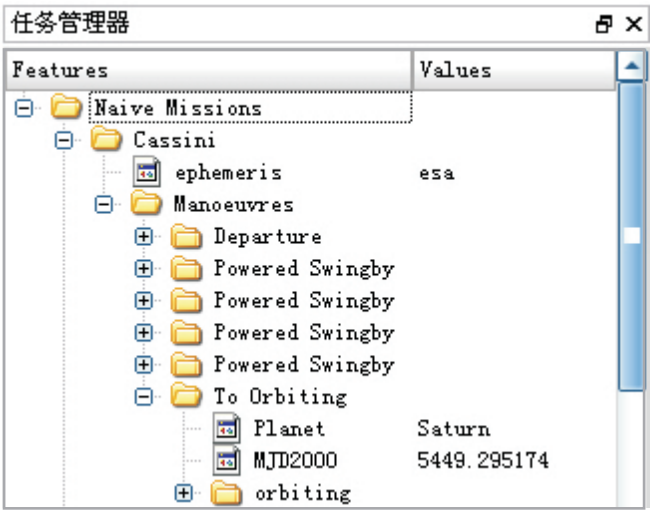


图 9－2　任务管理器截图

9.3.2　任务轨迹仿真

在上一节中已介绍，所有任务（Naive Mission 和 Integral Mission）都从虚任务基类 AbstractMission 继承，并实现获取任意时刻航天器在指定惯性坐标系中位置的接口 getPosVel。在轨迹仿真过程中，只需要在任务周期内以固定步长获取连续的离散轨迹，并在 OpenGL 中创建显示列表（Addison，2005）的方式进行线绘制。太阳系各天体的仿真则以纹理映射和 alpha 混合等技术实现真实感较强的 3D 仿真效果（图 9－3）。

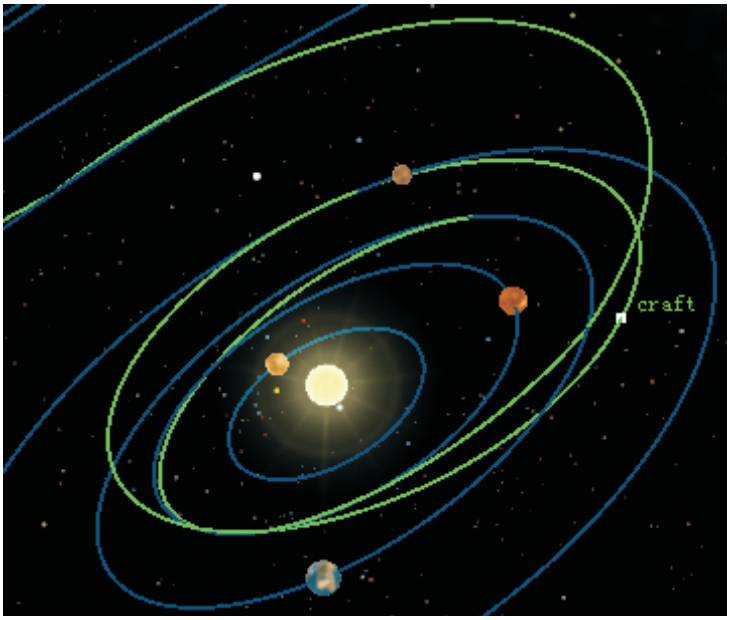


图 9－3　EVVEJS 轨迹仿真效果

9.3.3　星空背景仿真

常用的恒星光谱分类系统中（Keenan，1963；Morgan，*et al*，1973），哈佛系统是美国哈佛大学天文台于 19 世纪末提出的。这个系统的判据是光谱中的某些特征谱线和谱带，以及这些谱线和谱带的相对强度，同时也考虑连续谱的能量分布。本系统的光谱型用拉丁字母表示，组成如图 9－4 所示的序列。

各型之间是逐渐过渡的，每型又分为 10 个次型，用阿拉伯数字表示：O0，…，O9；B0，…，B9；…。这一序列由左到右，对应于温度的下降。最热的 O 型星温度约 40 000K，最冷的 M 型星约 3 000K。序列右端的 S、R 和 N 等分支则可能反映化学组成的差别。由于历史的原因，常把 O、B、A 型叫作早型，K、M 型叫作晚型，F、G 型叫作中型。各型星的颜色和

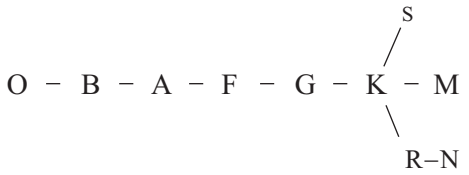


图 9-4 恒星光谱序列

在普通蓝紫波段的主要光谱特征描述见表 9-2。

表 9-2 恒星光谱类型和颜色对照表

光谱类型	可见光颜色	RGB(近似)
O 型	蓝白色	125 185 255
A 型	白色	255 255 255
F 型	黄白色	255 255 170
G 型	黄色	255 255 0
K 型	橙色	255 110 0
M 型	红色	190 50 0
R 和 N 型	橙到红色	190 50 0
S 型	红色	255 0 0

哈佛大学天文台于 1918—1924 年发表的《亨利·德雷伯星表》(HD 星表)载有 20 余万颗星的光谱类型,其中 99%的星属于 B～M 型,O、R、N、S 型很少,还有少数光谱不能归入上述序列,分别记为:P——行星状星云,W——沃尔夫-拉叶星。新星光谱曾记为 Q,但现在已不使用。因此在本书中,在进行星空背景仿真时,仅对 B～M 型光谱型的恒星进行了区分颜色的绘制,其他类型的恒星都在颜色上归类到了 M 型,即红色。

§9.4 任务优化模块

在§9.1 节中简要介绍了 AbstractOptimizer 类,在 Optimizer 模块中,可对任务参数、优化算法参数等参数信息进行修改。利用 PSO、DE 等优化算法(图 9-5 所示)实现对三脉冲变轨任务以及多、无借力飞行任务的优化仿真(图 9-6 所示)。

可通过 Mission Optimizer Monitor 利用 AbstractOptimizer 提供的控制接口对优化进程进行监控,优化进程监控界面见图 9-7。

图 9－5　设置任务优化参数

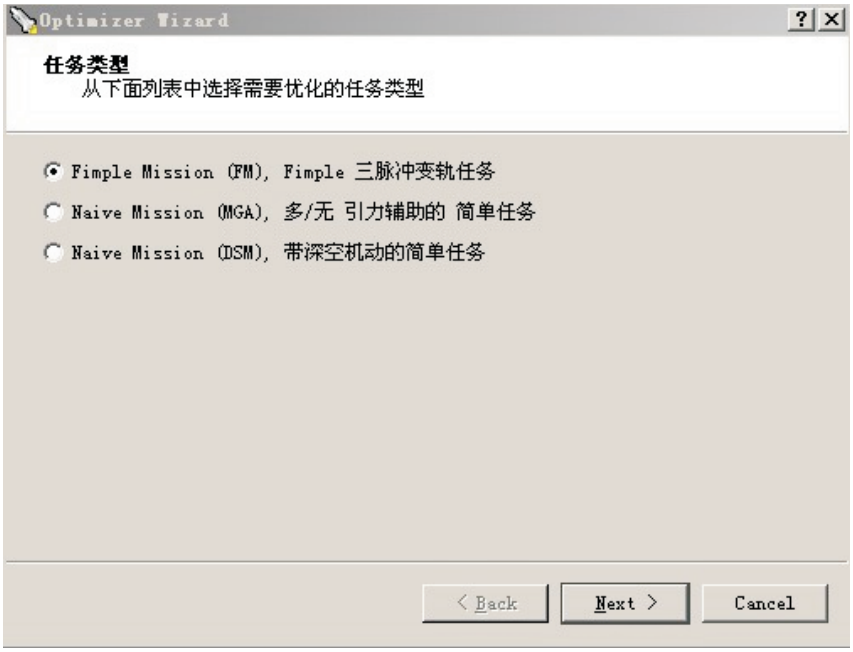


图 9－6　三脉冲变轨任务选择

图 9-7 优化进程监视器截图

附录 A　坐标旋转公式

直角坐标系中的矢量旋转公式：

绕 X 轴旋转 θ 度：$\begin{cases} x_{\text{new}} = x \\ y_{\text{new}} = y\cos\theta - z\sin\theta \\ z_{\text{new}} = z\cos\theta + y\sin\theta \end{cases}$

绕 Y 轴旋转 θ 度：$\begin{cases} x_{\text{new}} = x\cos\theta - z\sin\theta \\ y_{\text{new}} = y \\ z_{\text{new}} = z\cos\theta + x\sin\theta \end{cases}$

绕 Z 轴旋转 θ 度：$\begin{cases} x_{\text{new}} = x\cos\theta - y\sin\theta \\ y_{\text{new}} = y\cos\theta + x\sin\theta \\ z_{\text{new}} = z \end{cases}$

绕任意单位矢量 $\vec{r}$ 旋转 θ 度：

$$\begin{cases} x_{\text{new}} = [\cos\theta + (1-\cos\theta)r_x^2]x + \\ \qquad [(1-\cos\theta)r_x r_y - r_z\sin\theta]y + \\ \qquad [(1-\cos\theta)r_x r_z - r_y\sin\theta]z \\ y_{\text{new}} = [\cos\theta + (1-\cos\theta)r_y^2]y + \\ \qquad [(1-\cos\theta)r_x r_y + r_z\sin\theta]x + \\ \qquad [(1-\cos\theta)r_y r_z - r_x\sin\theta]z \\ z_{\text{new}} = [\cos\theta + (1-\cos\theta)r_z^2]z + \\ \qquad [(1-\cos\theta)r_x r_z - r_y\sin\theta]x + \\ \qquad [(1-\cos\theta)r_y r_z + r_x\sin\theta]y \end{cases}$$

附录 B 太阳系天体相关数据

天体	引力常数 (km^3/s^2)	轨道长半轴 (km)	轨道偏心率	赤道半径 (km)	影响球半径 (km)	轨道速度 (km/s)
太阳	1.327×10^{11}	—	—	6.96×10^5	1×10^{10}	—
月球	4.903×10^3	3.845×10^5	0.054 9	1.738×10^3	6.619×10^4	1.02
水星	2.203×10^4	5.791×10^7	0.205 63	2.44×10^3	1.124×10^5	47.83
金星	3.249×10^5	1.082×10^8	0.006 79	6.052×10^3	6.163×10^5	34.99
地球	3.986×10^5	1.496×10^8	0.016 72	6.378×10^3	9.246×10^5	29.76
火星	4.283×10^4	2.28×10^8	0.093 38	3.398×10^3	5.773×10^5	24.11
木星	1.267×10^8	7.784×10^8	0.048 45	7.14×10^4	4.821×10^7	13.05
土星	3.794×10^7	1.427×10^9	0.055 69	6.033×10^4	5.456×10^7	9.64
天王星	5.78×10^6	2.87×10^8	0.047 24	2.54×10^4	5.169×10^7	6.8
海王星	6.871×10^6	4.497×10^9	0.008 58	2.43×10^4	8.681×10^7	5.43

天文单位长度：$A_u = 1.495\ 978 \times 10^8$ km

万有引力常量：$G = 6.673 \times 10^{-11}\ m^3/(s^2 \cdot kg)$

参 考 文 献

陈宝林.最优化理论与算法[M].北京:清华大学出版社.1989

陈方杰,戴光明,彭雷等.Halo轨道及其附近不变流形[C].中国宇航学会深空探测技术专业委员会第四届学术会议.太原.2007

陈国良,王煦法,庄镇泉等.遗传算法及其应用[M].北京:人民邮电出版社.1996

崔晓峰.基于HLA的航天飞行任务联合仿真系统设计[J].飞行器测控学报.2005,24:21～26

戴光明,王茂才.多目标优化算法及在卫星星座设计中的应用[M].武汉:中国地质大学出版社.2009

房建成,宁晓琳.天文导航原理及应用[M].北京:北京航空航天大学出版社.2006

付兆萍,李红.轨道方程计算中Adams-Cowell方法的外推改进[J].中国空间科学技术.2006,(1):22～26

高飞,童恒庆.一类推广的差异演化算法及其应用[J].武汉大学学报(理学版).2005,51(5):49～56

耿长福.航天器动力学[M].北京:中国科学技术出版社.2006

龚胜平,李俊峰,高云峰等.基于不变流形的登月轨道设计[J].应用数学和力学.2007,28(2):183～190

谷立祥,刘竹生.使用遗传算法和B平面参数进行月球探测器地月转移轨道设计[J].导弹与航天运载技术.2003,(3):1～5

胡中波.差分演化算法及其在函数优化中的应用研究[D].武汉:武汉理工大学,2006

吉根林.遗传算法研究综述[J].计算机应用与软件.2004,21(2):69～72

蒋忠樟,成浩.一种求解多峰函数优化问题的演化算法[J].武汉大学学报(理学版).2006,52(3):335～339

孔锐睿,仇汝臣,周田惠.单纯形的加速算法[J].南京理工大学学报.2003,27(2):209～213

蓝朝桢,陈景伟,李建胜等.航天任务实时三维可视化仿真[J].测绘科学技术学报.2007,(24):47～50

李艳.单纯形算法在行星际轨道优化设计中的应用研究[D].武汉:中国地质大学(武汉).2009

刘林,胡松杰,王歆.航天动力学引论[M].南京:南京大学出版社.2005

刘林.航天器轨道理论[M].北京:国防工业出版社.2000

刘民广.差异演化算法及其改进[J].系统工程.2005,23(2):108～111

刘强,刘林.Adams-Cowell方法与KSG积分器的比较[J].紫金山天文台台刊.1998,17(1):20～29

刘暾,赵钧.空间飞行器动力学[M].哈尔滨:哈尔滨工业大学出版社.2003

刘勇,康立山,陈毓屏.非数值并行算法——遗传算法[M].北京:科学出版社,1998

罗治情,演化算法在行星际轨道优化设计中的应用研究[D].武汉:中国地质大学(武汉).2007

罗治情,戴光明,詹炜等.基于"优胜劣汰"原则的差异算子[J].计算机工程与设计.2004,27(16):2964～2965

罗治情,戴光明,郑蔚.稳定进化的遗传算法[J].计算机工程与应用.2004,21(2):72～74

马高峰,鲁强,郑勇.JPL行星/月球星历[C].中国宇航学会深空探测专业委员会第一届学术会议.哈尔滨.2005

马文臻.深空探测器轨道设计与优化[D].北京:中国科学院空间科学与应用研究中心.2006

马芸. 飞行轨迹产生软件仿真算法及应用 [J]. 导航. 2003(1):101～111
潘正君,康立山,陈毓屏.演化计算[M].北京:清华大学出版社. 2000
齐映红,曹喜滨. 三脉冲最优交会问题的解法[J]. 吉林大学学报(工学版). 2006,36(4):608～612
乔栋,崔祜涛,崔平远. 小行星探测最优两次脉冲交会轨道设计与分析[J]. 宇航学报. 2005a,26(3):362～367
乔栋,崔祜涛,崔平远等. 气动-借力飞行转移轨道研究及在行星探测中的应用[J]. 宇航学报. 2005,26(5):541～546
任远,崔平远,栾恩杰. 最优两脉冲行星际轨道转移优化算法[J]. 航空学报.2007,28(6):1307～1311
沈剑华. 数值计算基础[M]. 上海:同济大学出版社.1999
石再明,戴光明,罗治情. 行星际双脉冲转移轨道优化问题的复杂度分析[C]. 中国宇航学会深空探测技术专业委员会第三届学术会议. 西安.2006
石再明.行星际脉冲转移轨道设计与优化算法[D].武汉:中国地质大学(武汉).2008
王华. 交会对接仿真系统 [D]. 长沙:国防科技大学. 2002
王小平,曹立明.遗传算法——理论、应用与软件实现[M].西安:西安交通大学出版社.2002
王于. SQP 算法在轨道优化中的应用研究[D].武汉:中国地质大学(武汉).2009
王于.双脉冲卫星交会算法研究[D]. 武汉:中国地质大学(武汉).2006
吴涛,杨宜康,于军琪等. 飞行器轨道交会的 H∞次优控问题[J]. 中国空间科学技术. 2000,(6):36～42
郗晓宁,王威,高玉东. 近地航天器轨道基础[M]. 北京:国防科技大学出版社. 2003
席少霖.非线性最优化方法[M].北京:高等教育出版社.1992
向开恒,肖业伦. 空间交会中脉冲变轨燃料消耗研究[J]. 中国空间科学技术. 1999,(3):9～15
向礼杰. HLA 在深空探测仿真系统中的应用[D]. 哈尔滨:哈尔滨工业大学. 2006
肖伟,全惠云,史滋福.改进的遗传算法[J].计算机工程与应用.2004,40(4):53～54
谢晓锋,张文俊,张国瑞等.差异演化的实验研究[J].控制与决策.2004,19(1):49～56
熊伟清,刘明达,张少宇.一种具有性别特征的遗传算法[J].计算机工程.2005,31(1):165～186
徐成贤,陈志平,李乃成等.近代优化方法[M].北京:科学出版社.2002
徐晓云,李俊峰. 小卫星轨道姿态控制系统仿真软件平台 [J]. 清华大学学报(自然科学版). 2003,(2):234～237
杨嘉墀,范剑峰. 航天器轨道动力学与控制(上)[M]. 北京:中国宇航出版社. 2005
袁亚湘,孙文瑜.最优化理论与方法[M].北京:科学出版社.1997
张光澄.非线性最优化计算方法[M].北京:高等教育出版社.2005
张洪波,郑伟,汤国建. 混合遗传算法在远程交会轨道设计中的应用[J]. 航天控制. 2006,24(2):34～37
张然,乔栋,刘宇飞. 行星转移发射机会搜索算法研究[C]. 中国宇航学会深空探测技术专业委员会第五届学术会议. 长沙.2008
张占月,徐艳丽. 基于 STK 的航天任务仿真方案分析[J]. 装备指挥技术学院学报. 2006,17(1):48～51
郑光勇,曹俊彬,石武军.一个改进的非线性单纯形算法[J].邵阳学报.2006,3(2):18～19
郑亚弟. 导航载体轨迹仿真系统的研究与开发 [D]. 郑州:解放军信息工程大学. 2006
朱仁璋,蒙薇. 航天器交会两点边值问题[J]. 宇航学报. 2006,27(6):1182～1186

朱仁璋，尹艳，汤溢. 空间交会 N 次推力机动状态方程及控制算法[J]. 宇航学报. 2005,26(3):206～211

ACT. Astrodynamical routines[EB/OL]. http://www.esa.int/gsp/ACT/. 2008

Addison - Wesley 著. 邓郑详译. OpenGL 编程指南(第四版)[M]. 北京:人民邮电出版社,2005

Aerospace Industries, Inc. International technical services[EB/OL]. The Histroy of the Term C3. Dec. 2001

Bate R R 等著. 吴鹤鸣，李肇杰译. 航天动力学基础[M]. 北京:北京航空航天大学出版社. 1990

Becerra V M, Myatt D R, Nasuto S J, *et al*. An efficient pruning technique for the global optimisation of multiple gravity assist trajectories[J]. Proceedings of GO. 2005,1～7

Deb K, Agrawal S, Pratap A, *et al*. A fast elitist non - dominated sorting genetic algorithm for multi - objective optimization: NSGA - II[C]. Technical Report No. 2000001. Kanpur: Indian Institute of Technology Kanpur, India. 2000

Fred H, Walters Jr, Lloyd R, *et al*. Sequential simplex optimization boca raton[M]. Fla. CRC Press. 1991

Fukushima M. A success quadratic programming algorithm with global and superlinear convergence properties[C]. Mathematical Programming. 1986

George H B. Design of the approach trajectory: B-plane targeting[EB/OL]. ASEN 5519: Interplanetary Mission Design. 2005

Goldberg D E. Genetic algorithms in search, optimization and machine learning[M]. New York: Addison-Wesley Publishing Company, Inc. 1989

http://hyperphysics. phy - astr. gsu. edu/hbase/doubal. html. Double Ball Drop. 2007

http://www. mathpages. com/home/kmath114. htm. Gravitational Slingshot. 2007

http://www. online. tj. cn/dd99/teaching/2001a/jsjshuxue(2)/shuzhifenxi/10chff3. htm. 2006

Izzo D, Becerra V M, Myatt D R, *et al*. Search space pruning and global optimisation of multiple gravity assist spacecraft trajectories[J]. Jurnal of Global Optimisation manuscript. 2006

Kalyanmoy D, Amrit P, etc. A fast and elitist multi-objective genetic algorithm: NSGA - Ⅱ[J]. Evolutionary Computation. 1994,2(3):221～248

Kalyanmoy D, Nikhil P, Canesh N. Interplanetary trajectory optimization with swing-bys using evolutionary multi-objective optimization[C]. Second International Symposium, ISICA2007. 2007,9:26～35

Keenan P C. Classification of stellar Spectra, Basic Astronomical Data[M]. Univ. of Chicago Press, Chicago. 1963

Kennedy J, Eberhart R. Particle swarm optimization[C]. IEEE International Conference of Neural Networks, Perth, Australia. 1995

McCormick G P. Nonlinear programming: theroy, algorithms and placation[M]. John Wiley and Sons, New York. 1983

Morgan W W, Keenan P C. Spectral classification, annual review of astronomy and astrophysics[J]. Annual Reviews Inc. Palo Alta. 1973,11:29～50

Myatt D R, Becerra V M, Nasuto S J, *et al*. Advanced global optimisation tools for mission analysis and design[D]. 2003

Nelder J A, Mead R. A simplex method for function minimization[J]. Comput. J. 1965,7:308～313

Povoleril A, Lavagnal M, Finzil A E. Aero-gravity assist manoeuvres within preliminary interplanetary mission design: A multi-objective evolutive algorithm approach[C]. 18th International Symposium on Space Flight Dynamics. Haus der Bayerischen Wirtschaft, Munich, Germany. 2004

Sims J A. Delta-V gravity-assist trajectory design: theory and practice[D]. West Lafayette, US. Purdue University. 1996

Spendley W, Hext G R, Himsworth F R. Sequential application of simplex designs in optimization and evolutionary operation[J]. Techno Metrics. 1962,4:441～461

Srinivas N, Kalyanmoy D. Multiobjective optimization using nondominated sorting in genetic algorithms [J]. Evolutionary Computation. 1994

Srinivas N, Deb K. Multi - objective optimization using Non-dominated sorting in genetic algorithms[J]. Evolutionary Computation. 1994,2(3):221～248

Standish E M. Keplerian element for approximate positions of the major planets[EB/OL]. http://ssd.jpl.nasa.gov/? planet_pos. 2008

Storn R, Price K. Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces[J]. Global Optimization. 1997,11:341～359

Storn R, Price K. Differential evolution—a simple and efficient adaptive scheme for global optimization over continuous spaces[C]. Technical Report TR - 95 - 012, ICSI. 1995

Tamás V, Dario I, Claudio B. Benchmarking different global optimisation techniques for preliminary space trajectory design[C]. 58th International Astronautical Cngress, Hyderabad. 2007

Victor B C. Global optimisation and search space pruning in spacecraft trajectory design[J]. IEEE Colloquium on Optimisation for Control, Sheffield, UK. 24 April, 2006

Wolfram S. The mathematicia book[M]. Fifth Edition. Wolfram Media. 2003

策划编辑：段连秀
责任编辑：段连秀
封面设计：魏少雄

ISBN 978-7-5625-2804-3
9 787562 528043 >
定价:25.00元