

# 有向黑白旅行商问题

江 贺<sup>1),2)</sup> 张宪超<sup>1)</sup> 陈国良<sup>2)</sup>

<sup>1)</sup>(大连理工大学软件学院 大连 116621)

<sup>2)</sup>(中国科学技术大学计算机科学与技术系 合肥 230027)

**摘 要** 黑白旅行商问题是经典旅行商问题的推广,在基于 SONET 技术的光纤网络设计、航线调度等领域具有广泛的应用.已有研究工作集中在无向黑白旅行商问题上.文章研究该问题的更一般形式——有向黑白旅行商问题.首先,给出了有向黑白旅行商问题的混合整数线性规划公式.与目前无向黑白旅行商问题包含指数多个约束的规划公式相比,它仅包含多项式个约束.其次,给出了一种启发式算法.实验表明,该启发式算法能够有效地求解黑白旅行商问题的实例.由于无向黑白旅行商问题是有向黑白旅行商问题的特例,故文中的结论对于求解无向黑白旅行商问题同样有效.

**关键词** 有向黑白旅行商问题;NP-难解;混合整数线性规划;启发式算法  
**中图法分类号** TP301

## Directed Black and White Traveling Salesman Problem

JIANG He<sup>1),2)</sup> ZHANG Xian-Chao<sup>1)</sup> CHEN Guo-Liang<sup>2)</sup>

<sup>1)</sup>(School of Software, Dalian University of Technology, Dalian 116621)

<sup>2)</sup>(Department of Computer Science and Technology, University of Science and Technology of China, Hefei 230027)

**Abstract** The black and white salesman problem (BWTSP) is a generalization of classic traveling salesman problem (TSP) arising from work on the design of telecommunication fiber networks using the SONET technology, the scheduling of airline operation, etc. The existing research work focused on the undirected BWTSP. This paper investigates the more general form of this problem, i. e. the directed BWTSP. First, this paper proposes a mixed integer linear programming (MILP) formulation for the directed BWTSP. Compared to the formulation with exponential number of constraints for the undirected BWTSP, the new one contains polynomial number of constraints only. Furthermore, a heuristics for the directed BWTSP is presented. Experiments indicated that the heuristics could effectively solve instances of the BWTSP. Results of this paper could be also useful for the undirected BWTSP when viewed as a special case of the directed BWTSP.

**Keywords** directed black and white traveling salesman problem; NP-hard; mixed integer linear programming; heuristics

### 1 引 言

黑白旅行商问题 BWTSP(Black and White Trav-

eling Salesman Problem)最早由 Ghiani 等人<sup>[1-3]</sup>提出,在基于 SONET 技术的光纤网络设计<sup>[4]</sup>、航线调度<sup>[5]</sup>等领域具有广泛的应用背景.作为经典旅行商问题 TSP(Traveling Salesman Problem)的推广形

收稿日期:2005-03-18;修改稿收到日期:2006-06-05.本课题得到国家自然科学基金重大项目(90412007)、国家自然科学基金(60503003)、辽宁省自然科学基金(20051082)和大连理工大学青年教师培养基金资助.江 贺,男,1980年生,博士,讲师,研究方向为算法设计与分析、组合优化等. E-mail: jianghe@dlut.edu.cn. 张宪超,男,1971年生,博士,副教授,研究方向为算法设计与分析、组合优化、网络优化及应用. 陈国良,男,1938年生,教授,博士生导师,中国科学院院士,研究领域为并行算法、高性能计算及其应用.

式, BWTSP 问题属于典型的 NP-难解问题. 根据计算复杂性理论可知: 除非  $P=NP$ , 否则任何 NP-难解问题均不存在多项式时间的完全算法<sup>[6]</sup>. 故此, 目前研究有两个方向: 完全算法能保证得到最优解, 但是运行时间是指指数复杂度的; 启发式算法则仅要求找到近似解, 而在多项式时间内结束.

根据 BWTSP 问题中边是否对称, BWTSP 问题可以分成无向 BWTSP 问题和有向 BWTSP 问题两大类. 由于 BWTSP 问题提出时间较短, 已有的研究成果集中在无向 BWTSP 问题上. 在完全算法研究方面, Ghiani 等人<sup>[1]</sup>提出了无向 BWTSP 问题的混合整数线性规划公式, 该规划公式包含指数多个约束条件. 在启发式算法研究方面, Bourgeois 等人<sup>[2-3]</sup>将无向 BWTSP 问题的启发式算法划分为构造、可行化和优化等三个阶段, 提出了 C1-F-I, C2-F-I 和 C3-F-I 等三种不同的启发式算法, 并通过实验对它们的性能进行了比较.

本文研究 BWTSP 问题的更一般形式——有向 BWTSP 问题, 主要工作包括以下两点: (1) 提出了有向 BWTSP 问题的混合整数线性规划公式, 它对于设计分枝限界等完全算法具有重要意义. 原有无向 BWTSP 的规划公式利用了无向图的对称特性, 故难以推广到有向 BWTSP 问题. 本文在有向 BWTSP 问题的退化形式——非对称旅行商问题 ATSP (Asymmetric TSP) 的基础上, 提出了新的混合整数线性规划公式. 与无向 BWTSP 问题上包含指数多个约束条件的规划公式相比, 新规划公式仅仅包含多项式个约束条件. (2) 提出了有向 BWTSP 问题的启发式算法 LKH-BWTSP. 该算法通过定义惩罚函数的方式, 对目前求解 ATSP 的最好算法之一的 LKH 进行修改, 设计了新的初始解生成算子、局部搜索算子. 大量实验表明, 该启发式算法能够有效地求解 BWTSP 问题的实例.

本文第 2 节给出有向 BWTSP 问题的定义及相关知识; 第 3 节给出有向 BWTSP 问题的混合整数线性规划公式; 第 4 节提出求解有向 BWTSP 问题的启发式算法 LKH-BWTSP; 实验结果在第 5 节给出; 最后一节总结全文, 并指出今后的研究方向.

## 2 问题描述及相关定义

**定义 1.** 有序三重集合  $G=(V, A, c)$  称为一个有向加权图, 其中  $V$  为顶点集,  $A$  为弧 (有向边) 集,  $c$  为弧权矩阵 (任意一弧  $(i, j)$  的权值记为  $c_{ij}$ ).

$\forall V' \subseteq V$ , 记  $\delta_A(V') = \{(i, j) \mid (i, j) \in A, i \in V', j \in V'\}$ , 集合  $\delta_A(V')$  中弧的数目记为  $|\delta_A(V')|$ .

**定义 2.** 黑白有向加权图  $G_{BW}=(B \cup W, A, c)$  是顶点集合  $V=B \cup W$  的有向加权图, 其中  $B$  是黑色顶点集合,  $W$  是白色顶点集合,  $B \cap W = \emptyset$ .

**定义 3.** 黑白有向加权图  $G_{BW}$  中的一条 TSP 环路就是一条不重复地访问  $B \cup W$  中所有顶点的 Hamilton 环路, 记为  $t = \langle \pi(1), \pi(2), \dots, \pi(n) \rangle$ , 其中  $\pi$  为  $[1..n]$  上的排列, 记  $f(t) = c_{\pi(1)\pi(n)} + \sum_{1 \leq i \leq n-1} c_{\pi(i)\pi(i+1)}$  表示环路  $t$  的权值.

**定义 4.** 黑白有向加权图  $G_{BW}$  中一条 BWTSP 环路  $t_{BW}$  是满足以下两个约束条件的 TSP 环路: 1) 基数约束: 环路上连续的两个黑色顶点间的白色顶点数目不超过正整数  $Q$ , 其中  $Q$  被称为基数阈值; 2) 权值约束: 环路上连续的两个黑色顶点间路径的权值不超过正数  $L$ , 其中  $L$  被称为权值阈值.

**定义 5.** 记黑白有向加权图  $G_{BW}$  上全体 BWTSP 环路的集合为  $BWTSP(G_{BW})$ .  $G_{BW}$  上的有向 BWTSP 问题是寻求  $t_{BW}^* \in BWTSP(G_{BW})$ , s. t.  $f(t_{BW}^*) = \min_{t \in BWTSP(G_{BW})} f(t)$ , 其中  $t_{BW} \in BWTSP(G_{BW})$ .

**定义 6.** 黑白有向加权图  $G_{BW}$  中的一条有向 BB 路径是首尾为黑色顶点、内部为白色顶点的路径, 记为  $p = i_1 i_2 \dots i_k$ , 其中  $i_1, i_k \in B, i_2, i_3, \dots, i_{k-1} \in W$ , 且  $i_j \neq i_h (1 \leq j, h \leq k, j \neq h)$ . 记  $p$  上顶点的集合为  $V(p) = \{i_1, i_2, \dots, i_k\}$ ,  $|V(p)| = k$  表示  $p$  上顶点的个数; 记  $p$  上全体弧的集合为  $A(p) = \{(i_1, i_2), (i_2, i_3), \dots, (i_{k-1}, i_k)\}$ ,  $|A(p)| = k-1$  表示  $p$  上弧的数目,  $g(p) = c_{i_1 i_2} + c_{i_2 i_3} + \dots + c_{i_{k-1} i_k}$  表示路径  $p$  的权值.

**定义 7.** 黑白有向加权图  $G_{BW}$  上的任意一条 TSP 环路  $t$  可以分解为  $|B|$  条有向 BB 路径 (即  $t$  由这些有向 BB 路径首尾依次相连而得), 记该  $|B|$  条有向 BB 路径的集合为  $P(t)$ .

## 3 有向 BWTSP 问题的混合整数线性规划公式

本节给出有向 BWTSP 问题的混合整数线性规划公式, 它仅包含多项式个约束.

### 3.1 无基数约束及权值约束的有向 BWTSP 问题的规划公式

在不考虑基数约束及权值约束的前提下, 有向 BWTSP 问题退化为 ATSP 问题. ATSP 问题已存

在多种规划公式<sup>[7]</sup>,下面给出的是 MTZ 公式<sup>[8]</sup>:

$$\min \sum_{i \neq j} c_{ij} x_{ij} \quad (1)$$

$$\text{s. t. } \sum_{i \neq j} x_{ij} = 1, \quad \forall j \in V \quad (2)$$

$$\sum_{j \neq i} x_{ij} = 1, \quad \forall i \in V \quad (3)$$

$$x_{ij} = 0 \text{ 或 } 1, \quad (i, j) \in A \quad (4)$$

$$(n-1)x_{ij} + u_i \leq u_j + (n-2), \\ i, j \in V \setminus \{1\} \quad (5)$$

$$u_i \geq 0, \quad \forall i \in V \quad (6)$$

其中变量  $x_{ij}$  与弧  $(i, j)$  相关联,若  $x_{ij}=1$ ,则弧  $(i, j)$  出现在 TSP 环路中,否则不出现在 TSP 环路中. 约束式(2)、(3)和(4)保证了每个顶点的出度和入度均为 1(即解是由环路构成的).

变量  $u_i$  与顶点  $i$  相关联,它表明了顶点在环路中出现的顺序. 若有  $i, j \neq 1$  且  $x_{ij}=1$ ,则由约束式(5)可知:  $u_j \geq u_i + 1$ . 约束式(5)和式(6)被称为子环路删除约束,它们可以保证解不包含子环路,这一点容易由反证法得到:假设满足式(2)~(6)约束的某个解存在子环路,则该解至少包含两个子环路,且至少存在一个子环路  $t_1$  不包含顶点 1. 记  $t_1$  中包含的弧数目为  $|A(t_1)|$ ,则  $|A(t_1)| < n$  成立. 对于  $t_1$  上的任意一条弧  $(i, j)$ ,有不等式  $(n-1) + u_i \leq u_j + (n-2)$  成立(根据约束式(5)). 将  $t_1$  上所有  $|A(t_1)|$  条弧对应的不等式累加得到:  $(n-1)|A(t_1)| \leq (n-2) \cdot |A(t_1)|$ ,矛盾,故假设不成立.

### 3.2 基数约束

本小节考虑如何将基数约束加入到 3.1 节提出的规划公式中. 对于任意  $i \in W$ ,引入变量  $q_i$ ,它表示在访问  $i$  后环路还允许连续出现的白色顶点的个数. 基数约束可以通过如下方式分析:若  $i, j \in W$  且  $x_{ij}=1$ (即弧  $(i, j)$  出现在环路上),则  $q_j \leq q_i - 1$  成立,故得到约束:

$$S(1 - x_{ij}) \geq q_j - q_i + 1, \\ \forall (i, j) \in A, i \in W, j \in W \quad (7)$$

其中,  $S$  是一个常量,其取值应该足够大以保证式(7)成立,同时又尽可能小,使得约束尽可能紧致.  $q_j$  的最大值是  $Q-1$ ,而  $q_i$  的最小值是 0,从而  $q_j - q_i + 1$  最大可能值为  $Q-1-0+1=Q$ ,故可令  $S=Q$ ,从而将式(7)转化为

$$Q(1 - x_{ij}) \geq q_j - q_i + 1, \\ \forall (i, j) \in A, i \in W, j \in W \quad (8)$$

另外  $q_i$  的取值范围满足以下公式

$$0 \leq q_i \leq Q-1, \quad \forall i \in W \quad (9)$$

**定理 1.** 对于黑白有向加权图  $G_{BW}$  中任意 TSP 环路  $t$ ,式(8)和(9)是基数约束的充要条件.

证明. (1)充分性证明

反证法:假设存在某个满足式(8)和(9)的 TSP 环路  $t$  违反了基数约束,则  $t$  上至少存在一条有向 BB 路径  $p=i_1 i_2 \cdots i_k \in P(t)$ ,使得  $k-2 > Q$  成立. 由解  $t$  满足式(8),可知以下公式成立:

$$q_{i_2} \geq q_{i_3} + 1,$$

$$q_{i_3} \geq q_{i_4} + 1,$$

...

$$q_{i_{k-2}} \geq q_{i_{k-1}} + 1.$$

将这  $k-3$  个不等式累加得到:  $q_{i_2} \geq q_{i_{k-1}} + (k-3)$ ,再由  $k-2 > Q$  可知  $q_{i_2} \geq q_{i_{k-1}} + Q$ . 而另一方面由解  $t$  满足式(9)可知  $0 \leq q_{i_2}, q_{i_{k-1}} \leq Q-1$ ,从而  $q_{i_2} < q_{i_{k-1}} + Q$ ,矛盾,故假设不成立,得证.

(2)必要性证明

对于黑白有向加权图  $G_{BW}$  中满足基数约束的 TSP 环路  $t$  上的任意一条有向 BB 路径  $p=i_1 i_2 \cdots i_k \in P(t)$ ,显然有  $k-2 \leq Q$  成立. 为  $p$  上的每个白色顶点  $i_j (1 < j < k)$  构造  $q_{i_j} = k - j - 1$ ,此时容易验证式(8)和(9)均成立. 证毕.

式(8)和(9)构成了基数约束条件,其数目为  $|\delta_A(W)| + |W|$ .

### 3.3 权值约束

本小节考虑如何加入权值约束. 对于任意  $i \in BUW$ ,引入变量  $l_i$  表示环路允许从  $i$  至下一个黑色顶点之间路径的权值大小.

(1)若  $i \in BUW, j \in W$  且  $x_{ij}=1$ (即弧  $(i, j)$  出现在环路上),则  $l_j \leq l_i - c_{ij}$  成立,故得到约束:

$$M_{ij}(1 - x_{ij}) \geq l_j - l_i + c_{ij}, \\ \forall (i, j) \in A, i \in BUW, j \in W \quad (10)$$

其中,  $M_{ij}$  是一个常量,其取值应该足够大以保证式(10)成立,同时又尽可能小,使得约束尽可能紧致.  $l_j$  的最大值不超过  $L$ ,而  $l_i$  的最小值是 0,从而  $l_j - l_i + c_{ij}$  的值不超过  $L + c_{ij}$ ,故可令  $M_{ij} = L + c_{ij}$ ,从而将式(10)转化为

$$(L + c_{ij})(1 - x_{ij}) \geq l_j - l_i + c_{ij}, \\ \forall (i, j) \in A, i \in BUW, j \in W \quad (11)$$

(2)若  $i \in BUW, j \in B$  且  $x_{ij}=1$ ,则  $l_j \leq L, l_i \geq c_{ij}$  类似于式(11)的推导,得到约束:

$$l_i - c_{ij} \geq -(1 - x_{ij})c_{ij}, \\ \forall (i, j) \in A, i \in BUW, j \in B \quad (12)$$

另外,  $l_i$  的取值范围满足以下公式:

$$0 \leq l_i \leq L, \forall i \in B \cup W \quad (13)$$

**定理 2.** 对于黑白有向加权图  $G_{BW}$  中任意 TSP 环路  $t$ , 式(11)~(13)是权值约束的充要条件.

证明. (1)充分性证明.

反证法:假设存在某个满足式(11)~(13)的 TSP 环路  $t$  违反了权值约束, 则  $t$  上至少存在一条有向 BB 路径  $p = i_1 i_2 \cdots i_k$ , 使得  $\sum_{1 \leq r \leq k-1} c_{i_r i_{r+1}} > L$  成立. 由解  $t$  满足式(11), 可知以下公式成立

$$l_{i_1} \geq l_{i_2} + c_{i_1 i_2},$$

$$l_{i_2} \geq l_{i_3} + c_{i_2 i_3},$$

...

$$l_{i_{k-2}} \geq l_{i_{k-1}} + c_{i_{k-2} i_{k-1}}.$$

将这  $k-2$  个不等式累加得到  $l_{i_1} \geq l_{i_{k-1}} + \sum_{1 \leq r \leq k-2} c_{i_r i_{r+1}}$ , 再由式(12)可知  $l_{i_{k-1}} \geq c_{i_{k-1} i_k}$ , 从而有  $l_{i_1} \geq \sum_{1 \leq r \leq k-1} c_{i_r i_{r+1}} > L$  成立. 而另一方面由解  $t$  满足式(13)可知  $0 \leq l_{i_1} \leq L$ , 矛盾. 故假设不成立, 得证.

(2)必要性证明.

对于满足权值约束的 TSP 环路  $t$  上任意一条有向 BB 路径  $p = i_1 i_2 \cdots i_k$ , 显然有  $\sum_{1 \leq r \leq k-1} c_{i_r i_{r+1}} \leq L$  成立. 对于  $p$  上的顶点  $i_j$  ( $1 \leq j < k$ ), 令  $l_{i_j} = \sum_{j \leq r \leq k-1} c_{i_r i_{r+1}}$ . 容易验证式(11)~(13)均成立. 证毕. 式(11)~(13)构成了权值约束条件, 其数目为  $|A| + |B| + |W|$ .

## 4 有向 BWTSP 问题的启发式算法

由于有向 BWTSP 问题是 NP-难解, 故对大规模问题, 人们寻求在可接受的计算时间内能获得(在解的性能上)可接受解的启发式算法. 本节将给出一个求解有向 BWTSP 问题的启发式算法.

### 4.1 无基数约束及权值约束的有向 BWTSP 问题的启发式算法

在不考虑基数约束及权值约束的前提下, 有向 BWTSP 问题退化为 ATSP 问题. 与对称旅行商问题(STSP)相比, 目前 ATSP 问题的研究尚不充分, 其启发式算法大致分为三类: 环路构造算法, 如最近邻算法、贪心算法<sup>[9]</sup>等; 局部搜索算法, 如 KP 算法<sup>[10]</sup>、LKH 算法<sup>[11]</sup>、Ihper-k 算法<sup>[12]</sup>等; 基于最小圈覆盖的算法, 如 Patch 算法<sup>[13]</sup>、RA 算法<sup>[14]</sup>、COP 算法<sup>[15]</sup>等. 其中, LKH 算法是 Helsgaun 在经典

LK 算法的基础上提出的高效局部搜索算法, 是目前已知的求解 ATSP 问题最好的算法之一. LKH 算法对经典 LK 算法的改进主要是两方面: (1)用  $\alpha$ -nearness 启发集代替最近邻启发集; (2)使用更加先进的搜索规则. 算法 LKH 的基本框架如下: (1)生成 ATSP 问题的  $\alpha$ -nearness 启发集; (2)根据当前最好的两个解及  $\alpha$ -nearness 启发集来选择初始解; (3)在启发集上调用 LK 算子求得局部最优解; (4)若局部最优解优于当前最优解, 则令它为当前最优解, 并将其与此前最好解的公共边调整为对应顶点启发集的第一、第二位序; (5)重复步 2~4, 直至算法结束.

### 4.2 有向 BWTSP 问题的启发式算法

通过在 LKH 算法上添加惩罚函数, 本小节得到了求解带有基数约束及权值约束的有向 BWTSP 问题的 LKH-BWTSP 算法. 本小节首先给出惩罚函数的定义, 然后分别给出初始解的生成算子和局部搜索算子, 最后给出完整的 LKH-BWTSP 算法.

(1)惩罚函数的定义

对于黑白有向加权图  $G_{BW}$  上的任意一条 TSP 环路  $t = \langle \pi(1), \pi(2), \dots, \pi(n) \rangle$ , 定义它的带惩罚权值  $\varphi(t)$  如下:

$$\varphi(t) = f(t) + \alpha(t) + \beta(t) \quad (14)$$

其中  $f(t)$  是 TSP 环路  $t$  的权值.  $\alpha(t)$  表示环路  $t$  违反基数约束的惩罚函数,  $\beta(t)$  表示环路  $t$  违反权值约束的惩罚函数, 分别定义如下:

$$\alpha(t) = \sum_{p_i \in P(t)} [ |V(p_i)| > Q ] M_Q \quad (15)$$

$$\beta(t) = \sum_{p_i \in P(t)} \max(g(p_i) - L, 0) M_L \quad (16)$$

其中,  $M_Q, M_L$  分别表示预先设定的某个常数,  $[x] =$

$$\begin{cases} 1, & x = \text{true} \\ 0, & x = \text{false} \end{cases}, \max(x, y) = \begin{cases} x, & x \geq y \\ y, & x < y \end{cases}.$$

(2)初始解的生成算子 ChooseInitialTour

LKH-BWTSP 算法的初始解生成算子 ChooseInitialTour(如图 1 所示)是在经典 LKH 的初始解生成算子的基础上修改而成的, 其基本思路是通过构造首尾相连的有向 BB 路径来获取初始解: (1)一条有向 BB 路径的构造从某个黑色顶点开始, 在不违反基数约束和权值约束的情况下插入新的白色顶

① Gutin G, Zverovich A. Evaluation of the contract-or-patch heuristic for the asymmetric TSP. Manuscript, 2000. <http://citeseer.ist.psu.edu/gutin00evaluation.html>

点,使得新加入的边尽可能属于启发集并且属于当前最好的环路;(2)在有向 BB 路径违反基数约束或

权值约束的情况下,则试图加入黑色顶点,从而开始构造一条新的有向 BB 路径.

```

算子 ChooseInitialTour
输入: BWTSP 问题实例  $I$ 
输出: 初始解  $t_0$ 
begin
1. for every  $i \in BUW$  do  $flag(i) = FALSE$  //把所有的顶点标注为未被选择
2. 随机选择一个顶点  $i \in B$ , 令  $flag(i) = TRUE$  //选择一个黑色顶点
3.  $lc = 0, qc = 0$  //初始化当前正在构造的有向 BB 路径,  $lc$  代表当前路径当前的权值,
   //  $qc$  代表当前路径所含白色顶点数目
4. while 存在未被选择的顶点 do {
   if  $lc \leq L \ \&\& \ qc \leq Q$  //若当前正在构造的有向 BB 路径不违反基数约束和权值约束
   then {
   if  $\exists j \in W$  s. t.  $flag(j) = FALSE \ \&\& \ (i, j)$  属于启发集  $\ \&\& \ \alpha(i, j) = 0 \ \&\& \ (i, j)$  属于当前最好环路
   then {  $qc++$ ,  $lc = lc + c_{ij}$ ,  $i = j$ ,  $flag(i) = TRUE$  } //修改当前有向 BB 路径的信息
   else if  $\exists j \in W$  s. t.  $flag(j) = FALSE \ \&\& \ (i, j)$  属于启发集
   then {  $qc++$ ,  $lc = lc + c_{ij}$ ,  $i = j$ ,  $flag(i) = TRUE$  } //修改当前有向 BB 路径的信息
   else if  $\exists j \in W$  s. t.  $flag(j) = FALSE$ 
   then {  $qc++$ ,  $lc = lc + c_{ij}$ ,  $i = j$ ,  $flag(i) = TRUE$  } //修改当前有向 BB 路径的信息
   else { //取一个未被访问过的黑色顶点任取  $j \in B$  s. t.  $flag(j) = FALSE$ 
    $qc = 0, lc = 0, i = j, flag(i) = TRUE$  //开始一条新的有向 BB 路径的构造过程
   }
   } else { //若当前正在构造的有向 BB 路径已经违反基数约束或权值约束
   if  $\exists j \in B$  s. t.  $flag(j) = FALSE$  //如果有未选择的黑色顶点
   then {  $qc = 0, lc = 0, i = j, flag(i) = TRUE$  } //开始一条新的有向 BB 路径的构造过程
   else { //选择一个未曾访问的白色顶点
   if  $\exists j \in W$  s. t.  $flag(j) = FALSE$ 
   then {  $qc++$ ,  $lc = lc + c_{ij}$ ,  $i = j$ ,  $flag(i) = TRUE$  } //修改当前有向 BB 路径的信息
   }
   }
   }
5. return 由有向 BB 路径构成的初始解  $t_0$ 
end

```

图 1 算子 ChooseInitialTour

ChooseInitialTour 的时间复杂度分析如下: 步 1 的时间复杂度为  $O(n)$ ; 步 2、3 和 5 的时间复杂度均为  $O(1)$ ; 步 4 是一个次数为  $O(n)$  的循环, 每次循环内部的时间复杂度为  $O(n)$ . 综上可知, ChooseInitialTour 的时间复杂度为  $O(n^2)$ .

### (3) 局部搜索算子 LK-BWTSP

作为局部搜索算法的核心组成部分, 局部搜索算子的设计显著影响算法的总体性能. 本文对经典 LKH 算法中的局部搜索算子进行了修改, 得到求解 BWTSP 问题的 LK-BWTSP 算子(如图 2 所示). 它与原 LKH 算法的局部搜索算子的主要差别在于: 利用带惩罚权值  $\varphi(t)$  代替了原算子的正收益函数<sup>①</sup>. 为了叙述方便, 引入记号  $\vartheta$ : 对于黑白加权图  $G_{BW}$  上的任意一条 TSP 环路  $t = \langle \pi(1), \pi(2), \dots, \pi(n) \rangle$ ,  $\vartheta(t, S_1, S_2)$  表示用集合  $S_2$  中的边代替环路  $t$  上  $S_1$  中的边后得到的新环路, 其中  $S_1, S_2$  均表示边的集合, 且  $S_1 \cap S_2 = \emptyset$ . 对于  $i_1, i_2, \dots, i_k \in BUW$ , 记  $S(i_1, i_2, \dots, i_k) = \{(i_1, i_2), (i_3, i_4), \dots, (i_{k-1}, i_k)\}$ .

LK-BWTSP 的基本思想是: (1) 利用  $k$  ( $k = 2, 4, 6, 8, 10$ ) 条新边代替当前环路上的  $k$  条边得到新环路; (2) 若新环路的带惩罚权值小于当前环路,

则以新环路作为当前环路并转向(1), 否则直接转向(1). BWTSP-LK 的时间复杂度分析如下: 步 1 的时间复杂度为  $O(n)$ ; 步 2 的时间复杂度为  $O(2^5 d^4 n^2) = O(d^4 n^2)$ , 其中  $d$  表示启发集的规模; 步 3 的时间复杂度为  $O(1)$ . 故此, LK-BWTSP 算子的时间复杂度为  $O(d^4 n^2)$ .

### (4) LKH-BWTSP 算法.

LKH-BWTSP 算法的主体框架(如图 3 所示)是一个主循环, 每次循环由一系列步骤构成: (1) 调用 ChooseInitialTour 生成一条初始环路; (2) 调用局部搜索算子 LK-BWTSP 对初始环路进行优化, 获得当前解  $t$ ; (3) 若当前解的带惩罚权值比当前最好环路要好, 则将当前解保存为当前最好环路, 并调整启发集.

LKH-BWTSP 算法的时间复杂度分析如下: 算法步 1 和 3 的时间复杂度为  $O(1)$ ; 步 2 是次数为  $n$  的循环, 而步 2.1 的时间复杂度为  $O(n^2)$ , 步 2.2 的时间复杂度为  $O(d^4 n^2)$ , 步 2.3 的时间复杂度为

① 可以取  $M_Q, M_L$  的值为  $n \times \max(c_{ij})$  的倍数, 为了避免该方法对原 LKH 算法所能求解实例规模的影响, 当  $\max(c_{ij})$  或者  $n$  较大时, 可以适当按比例缩小  $c_{ij}$ .

$O(n)$ ,故每一次循环的时间复杂度为  $O(d^4 n^2) + O(n^2) + O(n) = O(d^4 n^2)$ ,步 2 的时间复杂度为

$O(d^4 n^3)$ . 综上可知, LKH-BWTSP 算法的时间复杂度为  $O(d^4 n^3)$ .

算子 LK-BWTSP

输入: BWTSP 问题实例  $I$ , 初始解  $t_0$

输出: 解  $t$

begin

1.  $t = t_0$

2. for every node  $i_1 \in B \cup W$  do

for  $i_1$  在环路  $t$  上的每一个相邻顶点  $i_2$  do

for  $i_2$  的启发集中每一个顶点  $i_3$  do

for  $i_3$  在环路  $t$  上的每一个相邻顶点  $i_4$  do

if  $\varphi(\vartheta(t, S(i_1, i_2, i_3, i_4), S(i_2, i_3, i_4, i_1))) < \varphi(t)$

then {  $t = \vartheta(t, S(i_1, i_2, i_3, i_4), S(i_2, i_3, i_4, i_1))$ , goto 2 }

else{

for  $i_4$  的启发集中每一个顶点  $i_5$  do

for  $i_5$  在环路  $t$  上的每一个相邻顶点  $i_6$  do

if  $\varphi(\vartheta(t, S(i_1, i_2, \dots, i_5, i_6), S(i_2, i_3, \dots, i_5, i_6, i_1))) < \varphi(t)$

then {  $t = \vartheta(t, S(i_1, i_2, \dots, i_5, i_6), S(i_2, i_3, \dots, i_5, i_6, i_1))$ , goto 2 }

else{

for  $i_6$  的启发集中每一个顶点  $i_7$  do

for  $i_7$  在环路  $t$  上的每一个相邻顶点  $i_8$  do

if  $\varphi(\vartheta(t, S(i_1, i_2, \dots, i_7, i_8), S(i_2, i_3, \dots, i_7, i_8, i_1))) < \varphi(t)$

then {  $t = \vartheta(t, S(i_1, i_2, \dots, i_7, i_8), S(i_2, i_3, \dots, i_7, i_8, i_1))$ , goto 2 }

else {

for  $i_8$  的启发集中每一个顶点  $i_9$  do

for  $i_9$  在环路  $t$  上的每一个相邻顶点  $i_{10}$  do

if  $\varphi(\vartheta(t, S(i_1, i_2, \dots, i_9, i_{10}), S(i_2, i_3, \dots, i_9, i_{10}, i_1))) < \varphi(t)$

then {  $t = \vartheta(t, S(i_1, i_2, \dots, i_9, i_{10}), S(i_2, i_3, \dots, i_9, i_{10}, i_1))$ , goto 2 }

}

}

}

}

3. return 解  $t$

end

图 2 算子 LK-BWTSP

算法 LKH-BWTSP

输入: BWTSP 问题实例  $I$

输出: 解  $t_{\text{best}}$

begin

1.  $len = +\infty$  // 保存最好环路的带惩罚权值

2. for  $i = 1$  to  $n$  do{

2.1  $t_0 = \text{ChooseInitialTour}()$ ; // 构造初始解

2.2  $t = \text{LK-BWTSP}(t_0)$ ; // 对初始解进行局部搜索, 获得当前解

2.3 if  $\varphi(t) < len$  // 如果当前解比目前最好解还要好, 则把当前解保存为最好解

then {

2.3.1  $len = \varphi(t)$

2.3.2  $t_{\text{best}} = t$  // 保存当前最好解

2.3.3 调整启发集 // 根据当前最好解调整启发集中边的位序

}

}

3. return 解  $t_{\text{best}}$

end

图 3 算法 LKH-BWTSP

## 5 实验及性能分析

目前, NP-难解问题启发式算法的性能评价包括 3 种方式: 最坏情况分析; 概率分析(平均情况分析); 实验分析。

最坏情况分析的目的在于刻划算法在最坏情况下所能保证解的性能。以退化的有向 BWTSP 问

题——ATSP 问题为例, 对于满足三角不等式的实例而言, 目前最坏情况下比值界最好的是 RA<sup>[14]</sup> 算法(它保证最终解的权值不超过全局最优解的  $\log_2 n$  倍), 而对于一般实例, 已经证明<sup>[16]</sup>: 如果  $P \neq NP$  且  $\sigma \geq 1$ , 不存在具有比值界  $\sigma$  的多项式时间启发式算法。

概率分析法是为了评价算法的平均性能。例如, 概率分析法可以用来估计一个启发式算法求解大规

模实例的误差范围. 概率分析法和最坏情况分析法均有不足: 这两种方法所涉及的数学公式都非常复杂, 并且所用实例与实际应用相差较大, 故分析结果对于求解实际的 BWTSP 问题意义不大. 目前, 研究人员通常采用实验分析方法来评价算法性能. 在实验分析中, 算法运行在一系列的实例上, 通过衡量算法在这些实例上的解来评价算法的性能.

本节首先在无向 BWTSP 实例上测试了 LKH-BWTSP, 并与目前求解无向 BWTSP 问题最好的算法 C3-F-I<sup>[3]</sup> 进行对比, 然后在有向 BWTSP 实例上测试了 LKH-BWTSP 算法的性能.

### 5.1 无向 BWTSP 实验

由于无向 BWTSP 问题是有向 BWTSP 问题的特例, 因此 LKH-BWTSP 算法同样可以用于求解无向 BWTSP 问题. 为了评价 LKH-BWTSP 算法的实验性能, 本文将其与目前最好的算法 C3-F-I 进行对比. 本文参照文献[3]中无向 BWTSP 问题实例的

生成方式, 按照以下方法生成实验所用实例.

(1) 在  $[0, 100]^2$  二维矩形内随机生成  $n - |B|$  个坐标为整数的白色顶点. 在  $[20, 80]^2$  或  $[40, 60]^2$  二维矩形内随机生成  $|B|$  个坐标为整数的黑色顶点, 其中取  $n = 100, 200, |B| = 0, 1, n$ .

(2) 边  $(i, j)$  的权值  $c_{ij} = \lfloor \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \rfloor$ .

(3) 令基数约束  $Q = \lfloor a |W| / |B| \rfloor$ , 其中  $a$  取 1.5 或 2.

(4) 为了确定权值约束  $L$ , 本文首先利用 LKH 算法求解本实例对应的 STSP 问题, 所得解的权值记为  $z$ . 本文令  $L = \lfloor bz / |B| \rfloor$ , 其中  $b$  取 2 或 2.5.

根据以上方法总共可以得到 16 个难度不同的测试实例.

本文实验环境是 P4 1.4GHz 微处理器, 256MB 内存, Redhat Linux 8.0 操作系统. 表 1、表 2 给出了实验结果, 其中  $n, |B|, Q, L$  分别表示实例的规模、黑色顶点数目、基数阈值、权值阈值;  $n_{\text{fea}}, t_{\text{min}}, t_{\text{mean}}$ ,

表 1 算法 LKH-BWTSP 在无向 BWTSP 实例上运行 10 次的实验结果

$n$	$ B $	黑色顶点分布	$z$	$Q$	$L$	$n_{\text{fea}}$	$t_{\text{min}}$	$t_{\text{mean}}$	$f_{\text{min}}$	$f_{\text{mean}}$
100	10	$[20, 80]^2$	730	14	146	10	79	113.4	786	793.4
				14	195	10	81	132.7	763	787.4
				18	146	10	60	98.3	754	761.2
		$[40, 60]^2$	710	14	142	10	75	184.5	932	954.0
				14	178	10	91	145.3	915	943.6
				18	142	10	88	138.6	907	925.4
200	20	$[20, 80]^2$	976	14	98	10	678	1245.1	1173	1202.6
				14	122	10	803	1298.6	1156	1176.3
				18	98	10	578	932.7	1143	1167.2
		$[40, 60]^2$	1062	14	122	10	520	610.2	1046	1123.1
				14	106	—	—	—	—	—
				18	133	7	1845	8763.1	1345	1416.1
18	106	—	—	—	—	—				
18	133	7	2843	9876.2	1315	1387.8				

表 2 算法 C3-F-I 在无向 BWTSP 实例上运行 10 次的实验结果

$n$	$ B $	黑色顶点分布	$z$	$Q$	$L$	$n_{\text{fea}}$	$t_{\text{min}}$	$t_{\text{mean}}$	$f_{\text{min}}$	$f_{\text{mean}}$
100	10	$[20, 80]^2$	730	14	146	10	48	92.2	802	811.2
				14	195	10	45	82.4	795	806.5
				18	146	10	36	86.1	786	810.3
		$[40, 60]^2$	710	14	142	10	45	84.4	978	1026.1
				14	178	10	37	78.6	965	1018.6
				18	142	10	46	88.7	936	978.1
200	20	$[20, 80]^2$	976	14	98	10	498	976.2	1263	1312.7
				14	122	10	365	902.7	1229	1289.4
				18	98	10	478	732.6	1256	1307.3
		$[40, 60]^2$	1062	14	122	10	420	510.1	1189	1231.2
				14	106	3	1503	6120.5	1573	1590.2
				14	133	9	1385	4964.3	1445	1527.3
18	106	3	2247	6780.1	1524	1538.8				
18	133	9	2081	5760.5	1423	1498.6				

$f_{\min}$ ,  $f_{\text{mean}}$  分别表示可行解的数目、最少运行时间、平均运行时间、可行解的最小权值、可行解的平均权值. 实验结果(表 1、表 2)表明, LKH-BWTSP 所获得的解的质量明显高于 C3-F-I, 这是因为 LKH 算法的优化性能远高于 C3-F-I 算法中的优化算子(简单 2-opt)的原因. 而从运行时间上看, C3-F-I 具有较大的优势, 原因是 LKH-BWTSP 的时间复杂度高于 C3-F-I. 在可行解的个数方面, 对于黑色顶点分布在  $[40, 60]^2$  上,  $n=200$  的情况, LKH-BWTSP 所获得可行解的个数少于 C3-F-I, 而两个算法在其他的实例上均获得了 10 个可行解. 原因在于: 对于容易的实例, 两个算法均可以得到可行解, 而对于较难的实例, 因为 LKH-BWTSP 算法没有专门进行可行化的操作, 故此其获得可行解的能力比 C3-F-I 要弱.

## 5.2 有向 BWTSP 实验

一般形式的有向 BWTSP 问题尚不存在广泛使用的测试实例库. 为了评价 LKH-BWTSP 算法的实验性能, 本小节采用类似无向 BWTSP 问题实例

的生成方法, 得到实验所需的 16 个有向 BWTSP 测试实例. 有向 BWTSP 实例与无向 BWTSP 实例生成方法的唯一差别是权值计算方法, 弧  $(i, j)$  的权值计算公式如下:

$$c_{ij} = \begin{cases} \lfloor \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \rfloor, & i < j \\ 0, & i = j \\ |x_i - x_j| + |y_i - y_j|, & i > j \end{cases}$$

本文在有向实例上测试了算法 LKH-BWTSP 的性能(每个实例分别运行 10 次, 令带惩罚权值  $\varphi(t)$  中参数  $M_Q=1000, M_L=100$ ). 表 3 给出了实验结果. 实验结果表明, LKH-BWTSP 对于大部分有向 BWTSP 实例均可以获得可行解, 且所获得可行解的质量比较稳定(可行解的最小值与平均值差别很小). 对于其中两个实例, LKH-BWTSP 算法不能获得可行解(表格中以“—”标注), 其中的原因可能是权值阈值过小的缘故(当权值阈值由 114 增加为 143 时, LKH-BWTSP 获得 9 个可行解).

表 3 算法 LKH-BWTSP 在有向 BWTSP 实例上运行 10 次的实验结果

$n$	$ B $	黑色顶点分布	$z$	$Q$	$L$	$n_{\text{fea}}$	$t_{\min}$	$t_{\text{mean}}$	$f_{\min}$	$f_{\text{mean}}$
100	10	$[20, 80]^2$	878	14	176	10	85	129.2	966	968.5
					220	10	72	143.7	954	962.2
					176	10	59	100.8	933	938.6
			18	220	10	61	77.7	915	917.3	
		$[40, 60]^2$	850	14	170	10	80	224.1	1089	1123.7
					213	10	97	185.3	1076	1084.8
	170			10	99	159.6	1080	1114.6		
	18	213	10	76	157.0	977	986.4			
200	20	$[20, 80]^2$	1145	14	115	10	982	1326.2	1234	1244.1
					144	10	1010	1444.8	1234	1241.8
					115	10	611	886.5	1215	1220.7
			18	144	10	477	623.9	1186	1197.5	
		$[40, 60]^2$	1140	14	114	—	—	—	—	—
					143	9	2254	10100.8	1552	1585.4
	114			—	—	—	—	—		
	18	143	9	3727	11728.6	1441	1468.4			

## 6 结论与展望

本文研究了有向 BWTSP 问题, 提出了仅包含多项式个约束条件的混合整数线性规划公式. 由于无向 BWTSP 问题可以非常容易转换为有向 BWTSP 问题, 故该规划公式也适用于无向 BWTSP 问题. 同时, 本文在有向 BWTSP 问题的退化问题 ATSP 的 LKH 算法基础上, 给出了求解一般形式的有向 BWTSP 问题的启发式算法 LKH-BWTSP. 实验结果表明, 该启发式算法可以有效地求解有向 BWTSP 问题的实例.

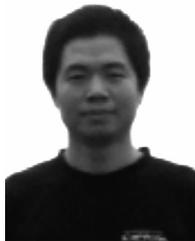
**致 谢** 感谢加拿大 Centre for Research on Transportation 的 Gilbert Laporte 教授为作者提供了 BWTSP 的相关资料. 同时特别感谢审稿人为完善本文所付出的辛勤工作!

## 参 考 文 献

- [1] Ghiani G, Laporte G, Semet F. The black and white traveling salesman problem. Centre for Research on Transportation, Montreal: Publication CRT-99-49, 1999
- [2] Bourgeois M, Laporte G, Semet F. Heuristics for the black and white travelling salesman problem. Centre for Research on Transportation, Montreal: Publication CRT-99-33, 1999

- [3] Bourgeois M, Laporte G, Semet F. Heuristics for the black and white travelling salesman problem. *Computers & Operations Research*, 2003, 30(1): 75-85
- [4] Soriano P, Wynants C, Seguin R, Labbe M, Gendreau M, Fortz B. Design and dimensioning of survivable SDH/SO-NET networks//Sanso B, Soriano P. *Telecommunications and Planning*. Boston: Kluwer, 1998: 147-167
- [5] Mak V, Boland N. Heuristic approaches to the asymmetric travelling salesman problem with replenishment arcs. *International Transactions in Operational Research*, 2000, 7(4): 431-447
- [6] Garey M R, Johnson D S. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. New York: W H Freeman and Company, 1979
- [7] Orman A J, Williams H P. A survey of different integer programming formulations of the travelling salesman problem. Department of Operational Research, London School of Economics and Political Science; LSEOR 04. 67, 2004
- [8] Miller C, Tucker A, Zemlin R. Integer programming formulations and travelling salesman problems. *Journal of ACM*, 1960, 7(2): 326-329
- [9] Bentley J L. Fast algorithms for geometric traveling salesman problems. *ORSA Journal Computing*, 1992, 4(4): 387-411
- [10] Kanellakis P C, Papadimitriou C H. Local search for the asymmetric traveling salesman problem. *Operational Research*, 1980, 28(5): 1006-1099
- [11] Helsingaun K. An effective implementation of the Lin-Kernighan traveling salesman heuristic. *European Journal of Operational Research*, 2000, 126(1): 106-130
- [12] Burke E K, Cowling P I, Keuthen R. Effective local and guided variable neighborhood search methods for the asymmetric travelling salesman problem//Boers E J W et al. *Proceedings of the EvoWorkshops 2001, Applications of Evolutionary Computing, Lecture Notes in Computer Science 2037*. Berlin: Springer-Verlag, 2001: 203-212
- [13] Karp R M, Steele J M. Probabilistic analysis of heuristics//Rinnooy A. H. G. etc. *The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization*. Wiley: Chichester, 1985
- [14] Frieze A, Galbiati G, Maffioli F. On the worst-case performance of some algorithms for the asymmetric traveling salesman problem. *Networks*, 1982, 12(1): 23-39
- [15] Glover F, Gutin G, Yeo A, Zverovich A. Construction heuristics and domination analysis for the asymmetric TSP. *European Journal of Operational Research*, 2001, 129(3): 555-568
- [16] Sahni S, Gonzales T. P-complete approximation algorithms. *Journal of the Association for Computing Machinery*, 1976, 23(3): 555-565

**JIANG He**, born in 1980, Ph. D., lecturer. His research interests include algorithm design and analysis, combinatorial optimization.



professor. His research interests include algorithm design and analysis, combinatorial optimization, network optimization and its applications.

**CHEN Guo-Liang**, born in 1938, professor, Ph. D. supervisor, member of Chinese Academy of Science. His research interests include parallel algorithms, high performance computing and its applications.

**ZHANG Xian-Chao**, born in 1971, Ph. D., associate

## Background

This project is supported by the Major Program of National Natural Science Foundation of China (90412007), the National Natural Science Foundation of China (60503003), the Natural Science Foundation of Liaoning Province (20051082), and the Gifted Young Foundation of Dalian University of Technology. It focuses on the efficient heuristics for classic NP-Hard problems (such as SAT, TSP, Bin-Packing and Max-Flow).

The group has presented some fast algorithms and theoretic results of NP-Hard problems and published some papers on *Journal of Combinatorial Optimization*, *Discrete Mathematics*, *Information Processing Letters*, *Journal of Computer Science and Technology*, *Journal of Advanced Software*, *Chinese Journal of Computers*(in Chinese), *Journal of Software* (in Chinese), etc. And being a part of this project, this paper mainly focuses on the heuristic algorithm design of TSP.