

文章编号: 1001-0920(2007)12-1341-06

奖励收集斯坦利最小树的混合拉格朗日与分散搜索算法

潘常春, 杨根科

(上海交通大学 自动化系, 上海 200240)

摘要: 针对 PCSTP 问题, 提出了 HL GSS 混合算法. 通过拉格朗日松弛策略, 将 PCSTP 问题转化为简单的 CMST 问题; 然后由 Volume 算法求解 PCSTP 的拉格朗日对偶问题并获得其下界. 用 SS 算法优化原问题的可行解, 利用求解拉格朗日对偶问题过程中获得的原始-对偶信息来指导 SS 算法的搜索. 仿真结果表明, HL GSS 比 SS 降低了算法的搜索空间, 加速了算法的收敛性.

关键词: 奖励收集斯坦利最小树; 拉格朗日松弛; 分散搜索; 混合算法

中图分类号: O232 **文献标识码:** A

Hybrid Lagrange and scatter-search algorithm for prize-collecting Steiner tree problem

PAN Chang-chun, YANG Gen-ke

(Department of Automation Engineering, Shanghai Jiaotong University, Shanghai 200240, China. Correspondent: PAN Chang-chun, E-mail: pan-cc@sjtu.edu.cn)

Abstract: The HL GSS is proposed for the PCSTP. The problem is transformed into an easier tractable CMST problem by using the method of Lagrange relaxation. A lower bound is obtained by solving the Lagrange dual problem and using the Volume algorithm. Then, the SS algorithm is used to optimize the feasible solutions of the PCSTP. The primal-dual information produced during solving the Lagrange-dual problem is exploited to guard the SS algorithm in the underlying procedure. The simulation results show that the proposed algorithm intensifies the search in the most promising regions of the solution space and thus speeds up the search time comparing with the SS algorithm.

Key words: Prize-collecting Steiner tree problem; Lagrange relaxation; Scatter search; Hybrid algorithm

1 引言

奖励收集斯坦利最小树问题(PCSTP)是组合优化问题研究的热点问题之一. 很多实际问题(比如有线电视线路的铺设、电信光缆的布局、城市公交站点的安排等)都可以抽象为此类问题. 从理论上讲, PCSTP 是 Steiner 最小树问题(STP)的一种泛化. 因为 STP 是 NP-hard 问题^[1,2], 所以 PCSTP 也必定是 NP-hard 问题. 多年来, 研究者做了大量关于 PCSTP 的研究^[1,3,4]. 一般讲, 对于此类问题主要有 3 种方法: 1) 数学规划方法; 2) 启发式方法; 3) 混合算法. 数学规划方法是基于精确的数学模型的求解, 能够得到问题的最优解或比较接近最优解的次优解, 但需要耗费大量的时间, 这对于大规模问题是难以接受的. 而启发式方法则比较灵活, 求解速度快, 但对于解的质量如何无法判断, 类似于黑箱操作, 这

就需要对解的质量和运行时间进行折衷. 相对于上述方法, 混合算法^[4,5]显示出了很大的优越性, 它能综合多种算法的优点, 获得高质量的解.

本文提出的算法就是通过数学分析深入了解问题本质, 得到比原始信息更为丰富的信息, 称之为二次信息. 在综合二次信息和原始信息的基础上, 采用启发式方法求解问题, 所得到的解将比直接在原始信息的基础上采用启发式方法所得到的解好得多. 更重要的是, 在大部分情况下, 二次信息能为启发式算法提供更为明确的结束准则或更为有效的搜索域, 而启发式方法则是以达到一定的运行代数后而结束的.

本文针对 PCSTP, 提出了基于拉格朗日松弛与分散搜索混合算法(HL GSS). 通过问题的分解-合成, 利用求解拉格朗日对偶问题过程中获得的原始

收稿日期: 2006-08-28; 修回日期: 2006-11-28.

基金项目: 国家自然科学基金项目(60574063).

作者简介: 潘常春(1979—), 男, 南京人, 博士生, 从事生产过程调度与优化的研究; 杨根科(1963—), 男, 山西平原人, 教授, 博士生导师, 从事混合系统控制、供应链与网络协议分析的研究.

对偶信息来寻求问题的最优解.从方法论的角度看,这种方法能够很方便地推广到其他类型的组合优化问题.

2 数学模型描述

考虑 PCSTP 中的一种形式:给定限额问题 (Quota problem)^[1].给定一个连通的无向图

$$G = (V, E).$$

其中: $V = \{1, 2, \dots, n\}$ 是节点集; E 是对应的弧集, 每一条弧 $e \in E$ 都对应一个权 C_e . 通常认为 G 是连通的, 每一个节点 $i \in V$ 都对应于一个非负的奖励 $p_i \geq 0$. 不失一般性, 设节点 1 为根节点 (Root node), 且认为 $p_1 = 0$. 现在给定奖励额度 Q , 目标是在无向图 G 中寻求一子树 $T = (S, E(S))$, 其中 $S \subseteq V$ 为节点集, $E(S) \subseteq E$ 为构成的子树的弧集, 且满足条件: 根节点 1 $\in S$, $\sum_{j \in S} p_j \leq Q$, 使得子树 T 中的所有弧的权重之和最小, 即 $\minimize \sum_{e \in E(S)} C_e$.

为了简化问题, PCSTP 可以方便地转换为约束最小生成树 (CMST) 问题^[4], 只要在原图 G 中引入一个哑节点 0, 得到一个增广图 $\bar{G} = (\bar{V}, \bar{E})$. 其中: $\bar{V} = V \cup \{0\}$, $\bar{E} = E \cup \{e\{0, j\}, j \in V\}$. 为了使 PCSTP 的每一个可行解都对应 \bar{G} 中的一个生成树, 对 \bar{G} 中的生成树定义如下约束:

约束 1 定义哑节点 0 到其他实际节点的权为 0, 即 $C\{0, j\} = 0, j \in V$. 为了与哑节点对应, 这里将原节点称作实际节点.

约束 2 在生成树中, 弧 $e\{0, 1\}$ 始终属于构成最小生成树的弧集合. 因为 1 是根节点, 必须将它包括在内.

约束 3 在生成树中, 除根节点 1 之外的所有实际节点, 即 $j \in V \setminus \{1\}$, 一旦与哑节点 0 建立了连接, 就不能与其他节点建立连接.

约束 4 与哑节点 0 连接的实际节点的奖励的总和不能超过 $\bar{Q}, \bar{Q} = \sum_{i \in V} p_i - Q$. 这对应了 PCSTP 中访问实际节点的奖励总和必须大于 Q .

图 1 形象地描述了 PCSTP 中如何将一个原图 G 中的子树转换为增广图 \bar{G} 中的一个受约束的生成树 (图 1 中虚线表示扩充的弧). 因此, 任意一个满足

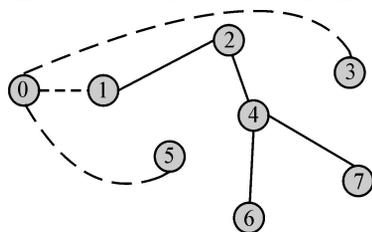


图 1 PC-STP 问题转化为 CMST 问题

约束的 \bar{G} 中的生成树都由两部分组成: 一部分是由实际节点构成的子树 (如图 1 中 1, 2, 4, 6, 7), 这部分对应了原 PCSTP 问题的一个可行解; 另一部分由哑节点 0 和没有被选中的实际节点构成子树 (如图 1 中 0, 3, 5). 两个子树通过弧 $e\{0, 1\}$ 连接.

由此可建立 CMST 问题的线性整数规划模型: 如果弧 $\{i, j\}$ 存在于最优解中, 则取自变量 $x_{\{i, j\}} = 1$; 否则 $x_{\{i, j\}} = 0$. 优化目标

$$Z_{CMSTP} = \min \sum_{\{i, j\} \in \bar{E}} C_{\{i, j\}} x_{\{i, j\}}. \quad (1)$$

s. t.

$$x_{\{0, 1\}} = 1; \quad (2)$$

$$x_{\{0, j\}} + x_{\{i, j\}} \leq 1, \quad \forall j \in V \setminus \{1\}, \{i, j\} \in \bar{E}; \quad (3)$$

$$\sum_{j \in V \setminus \{1\}} x_{\{0, j\}} p_j \leq \bar{Q}; \quad (4)$$

$$x_{\{i, j\}} \in \{0, 1\}, \text{ 且 } x = \{x_{\{i, j\}}\} \text{ 是定义在 } \bar{G} \text{ 中的一个生成树.} \quad (5)$$

其中: 式 (1) 为目标; 式 (2) ~ (4) 分别对应上述约束 2 ~ 约束 4.

3 拉格朗日松弛及其对偶算法

由式 (1) ~ (5) 可知, 约束 3 和约束 4 属于复杂约束, 如果能将它们松弛, 则原问题就变得简单了.

分别对约束集式 (3) 和 (4) 引入拉格朗日算子 $\lambda_{\{i, j\}} \geq 0 (j \in V \setminus \{1\}, \{i, j\} \in \bar{E})$ 和 $\mu \geq 0$, 得到如下拉格朗日函数:

$$Z_{LB}(\lambda, \mu) = \min \sum_{\{i, j\} \in \bar{E}} C_{\{i, j\}} x_{\{i, j\}} + \sum_{\{i, j\} \in \bar{E}, j \in V \setminus \{1\}} \lambda_{\{i, j\}} (x_{\{0, j\}} + x_{\{i, j\}} - 1) + \mu \left(\sum_{j \in V \setminus \{1\}} x_{\{0, j\}} p_j - \bar{Q} \right), \quad (6)$$

s. t. 式 (2), (5).

化简得

$$Z_{LB}(\lambda, \mu) = \min \sum_{\{i, j\} \in \bar{E}} \bar{C}_{\{i, j\}} x_{\{i, j\}} + K(\lambda, \mu), \quad (7)$$

s. t. 式 (2), (5).

式中

$$\bar{C}_{\{i, j\}} = C_{\{i, j\}} + \lambda_{\{i, j\}} + \lambda_{\{i, j\}}, \quad \forall \{i, j\} \in \bar{E} \setminus \{1\};$$

$$\bar{C}_{\{1, j\}} = C_{\{1, j\}} + \lambda_{\{1, j\}}, \quad \forall \{1, j\} \in \bar{E} \setminus \{1\};$$

$$\bar{C}_{\{0, j\}} = p_j + \sum_{\{i, j\} \in \bar{E} \setminus \{0, j\}} \lambda_{\{i, j\}}, \quad \forall j \in V \setminus \{1\};$$

$$K(\lambda, \mu) = - \sum_{j \in V \setminus \{1\}} \sum_{\{i, j\} \in \bar{E} \setminus \{0, j\}} \lambda_{\{i, j\}} - \mu \bar{Q}.$$

其中: $x = \{x_{\{i, j\}}\}, j \in V \setminus \{1\}, \{i, j\} \in \bar{E}$ 为矢量, $\bar{E}(j)$ 是与节点 j 相连的所有弧的集合.

显然, 给定 (λ, μ) 之后, 问题 (7) 已变成一个定义在 \bar{G} 中的简化了的 CMST 问题, 且 $Z_{LB}(\lambda, \mu)$ 是

Z_{CMSTP} 的一个下界. 由于此时的 CMST 问题的约束已经大大简化, 如果不考虑约束 (2), 它就是一个最小生成树 (MST) 问题. 对于 MST 问题, 采用多项式算法可以解决, 如 Prim's 算法. 首先将弧 $e \in \{0, 1\}$ 添加到所选的树中, 即可使用 Prim's 算法求解余下的 MST 问题.

为能快速产生 PCSTP 问题的可行解, 本文根据文献 [4] 提出的 MPA 算法, 进行了一些改进, 添加了一个自适应的系数 α . 的用途是为了能在算法早期更多地考虑奖励大的节点, 而在算法后期更多地考虑弧的权重. 改进后的 MPA 算法步骤如下:

Step1: 初始化

$$S = \{1\}; q = 0; U = V \setminus \{1\}.$$

Step2: 节点选择策略

While $q < Q$ do

1) 对所有的 $j \in U$, 计算比例系数

$$f_j = \frac{\min_{i \in S} \{C_{i,j}\}}{(p_j)}.$$

其中: 定义 $\alpha = (Q - q) / Q$, 该参数用来描述奖励 $p_j(j \in U)$ 对于所选择节点的影响程度, 根据定义可以确定 $0 < \alpha < 1$.

2) 取 $k = \arg(\min_{j \in U} f_j)$.

3) 取 $S = S \cup \{k\}, q = q + p_k, U = U \setminus \{k\}$.

End(while).

Step3: 构造最小树

用 Prim's 算法求解定义在节点子集 S 及其对应弧集合中的 MST 问题, 得到树 T .

Step4: 节点删除策略

1) 取 $P = \{j \in S \setminus \{1\} : q - p_j < Q \text{ 且 } \deg(j) = 1\}$, 其中 $\deg(j)$ 表示节点 j 的度. 对于每一个 $j \in P$, 定义 w_j 为树 T 中连接到节点 j 的弧的权重.

2) 如果存在 $P = \emptyset$, 则算法停止; 否则, 令 $S = S \setminus \{k\}$, 其中 $k = \arg(\max_{j \in P} w_j)$.

容易证明, MPA 算法的复杂度与 Prim's 复杂度相同.

对于任意一组可行的 Lagrange 乘子 (λ, μ) , $Z_{LB}(\lambda, \mu)$ 总是原问题的一个下界, 这里希望能够找到接近最优解的下界. 定义如下的 Lagrange 对偶问题:

$$Z_{LD} = \max_{\lambda, \mu} (Z_{LB}(\lambda, \mu)).$$

采用文献 [6] 提出的 Volume 算法来解决上述 Lagrange 对偶问题. 这种算法是次梯度法的一种延伸, 与典型的次梯度法相比, 具有以下优点:

- 1) 能够同时给出原始问题和对偶问题的解;
- 2) 定义了更为明确的算法停止准则.

Volume 算法已成功应用于多种组合优化问题, 如集分割问题、集覆盖问题和最大割问题^[7] 等. 下面给出具体流程.

Step1: 初始化

1) 令 $\bar{\lambda} = [\lambda^0, \mu^0]^T$, 其中 $[\lambda^0, \mu^0]$ 为可行的 Lagrange 向量乘子.

2) 解 $Z_{LB}(\bar{\lambda}^0, \mu^0)$, 令 \bar{x} 为当前问题的最优解, $\bar{z} = Z_{LB}(\bar{\lambda}^0, \mu^0)$.

3) 令 $t = 1$.

Step2:

1) 计算次梯度向量 $\bar{v} = [\bar{v}^1, \bar{v}^2]$, 其中

$$\bar{v}_{i,j}^1 = \bar{x}_{i,j} + x_{i,j} - 1, \quad \forall j \in V \setminus \{1\}, \{i, j\} \in V;$$

$$\bar{v}_{j \in V \setminus \{1\}}^2 = \bar{x}_{j \in V \setminus \{1\}} p_j - \bar{Q}.$$

2) 计算步长 α^t , 令

$$\alpha^t = \frac{U - \bar{z}}{v^2}.$$

其中: $\alpha^t \in [0, 2]$ 为步长系数, U 为最优的目标值. 采用人工上界的方法逼近最优目标值, U 被初始化为 $U = 1.05 * Z_{LB}(\bar{\lambda}^0, \mu^0)$.

3) 令 $\bar{\lambda}^t = \bar{\lambda}^0 + \alpha^t \bar{v}^1$, 并令 $[\lambda^t, \mu^t] = \bar{\lambda}^t$.

4) 解 $Z_{LB}(\lambda^t, \mu^t)$, 令 x^t 为当前问题的最优解, 并令 $z^t = Z_{LB}(\lambda^t, \mu^t)$.

5) 令 $v^t = [v^t, v^t]$, 其中

$$v_{i,j}^1 = x_{i,j}^t + x_{i,j}^t - 1,$$

$$v_{j \in V \setminus \{1\}}^2 = x_{j \in V \setminus \{1\}}^t p_j - \bar{Q}.$$

Step3:

1) 计算系数 α^t (用以估计原始问题的最优解), 令 $\alpha^t_{opt} = \arg(\min_{\alpha} (v^t + (1 - \alpha) \bar{v}^2))$.

2) 如果 $\alpha^t_{opt} < 0$, 则令 $\alpha^t = \alpha^t_{max} / 10$; 否则, 令 $\alpha^t = \min(\alpha^t_{max}, \alpha^t_{opt})$.

3) 令 $\bar{x}^t = \alpha^t x^t + (1 - \alpha^t) \bar{x}$.

Step4: 如果 $z^t > \bar{z}$, 则令 $\bar{z} = z^t, \bar{\lambda} = \lambda^t$.

Step5: 如果连续 300 代运算都没有当前最优解获得提升, 则算法停止; 否则令 $t = t + 1$, 返回 Step1.

注 1 在运算迭代过程中, 当前最优解超过人工上界 $0.95U$ 时, 增加人工上界 $U = U * (1 + 5\%)$.

注 2 本文中, 当 $t = 1$ 时取 $\alpha^t = 0.1$. 如果 $z^t < \bar{z}$ 保持连续 50 次迭代过程, 且 $\alpha^t > 10^{-4}$, 则令 $\alpha^t = *0.67$; 如果出现 $z^t > \bar{z}$, 且 $\bar{v} \cdot v^t < 0, \alpha^t < 1$, 则令 $\alpha^t = *2$. 取 α^t_{max} 初始值为 0.001, 每隔 100 代, 计算 \bar{z} 的增长幅度, 若其增长幅度小于 1%, 且 $\alpha^t_{max} > 10^{-5}$, 则 α^t_{opt} 减半^[4,7].

4 混合拉格朗日与分散搜索算法

需要指出的是,根据问题的定义,对于未选中的节点没有加惩罚,所以在大多数情况下,通过 Volume 算法求解 Lagrangian 对偶问题得到的最终的收敛解大都是不可行解(如果得到的是可行解,则表明得到了原有问题的最优解),即只是原问题的一个下界.通常的方法是将获得的下界作为一个部分解,通过一系列的修补策略使得这个不可行解变为可行解^[8].这种方法只用到了最后的下界信息,而抛弃了在求解 Lagrange 对偶问题过程中获得的大量信息.文献[9]证明,TSP 的最优解所包含的弧一定包含在求解 Lagrange 对偶问题过程中获得的弧集中,而且对于 MST 问题有同样的结果.所以通过保留求解 Lagrange 对偶问题过程中得到的弧集合,本文认为对于最优解的搜索可以限制在这个范围内^[2,4].将所有构成这些 MST 问题解的这些弧集合用 E 表示,这里只保留弧,不必保留求解 Lagrange 对偶问题过程中弧对应的权 $\bar{c}_{i,j}$,因为每次迭代弧的权都是随着 Lagrange 乘子的改变而改变的.本文提出的 HL GSS 算法就是为了利用这个特性. HL GSS 的流程如图 2 所示.

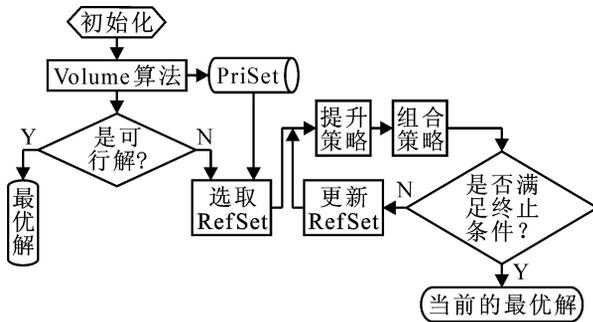


图 2 HL GSS 流程

SS 算法^[10]是一种基于启发式的进化算法,最早由 Glover 于 1977 年提出,已成功应用于多种优化问题,如图着色问题、并行机问题和 VRPTW 问题等^[11-13].其主要思想是:构建-提升-合成.构建策略就是通过一些启发式方法产生初始的参考解集,用 RefSet 表示,一般不超过 20 个,这些解不仅要求质量高,而且还要兼有多样化;提升策略是指对参考集中的解采取局部搜索策略进行处理,以提高解的质量且保持解的多样性;合成策略是指通过策略挑选参考解集中的一些子集,将这些子集通过合成机制组合为一个解,并用最后产生的新解来更新 RefSet;然后根据更新的 RefSet 进行新一次搜索.

由此可见,对于 SS 算法,初始解的选择是很重要的.在求解 Lagrange 对偶函数的过程中,对给定的一组 Lagrange 乘子,每次迭代都会改变弧的权

重,构成一个新图 $G = (V, E)$.其中 t 为对应的迭代次数,当前的弧权的计算见式(7).利用 MPA 算法作用于 G 上,可获得一个可行解,随着 Lagrange 乘子的不断更新,会产生一系列不同的连通图.不必每代都采用 MPA 求解,而是每隔若干代利用 MPA 作用于 G 产生一个可行解,并保留在可行解集中,用 PriSet 表示;然后从 PriSet 中取大约 10 个质量最好的解作为 SS 算法的 RefSet.后续的 SS 算法搜索的空间将限定在简化图 $\tilde{G} = (V, \tilde{E})$ 内,令此时弧的权为 $\tilde{c}_e, e \in \tilde{E}$ 仍保持为原来的权 c_e .选取策略可参见文献[10], Tree-Tabu-Search (TTS) 局部搜索策略^[14]可应用问题的提升策略,组合策略和 SS 算法框架实现可参见文献[13].限于篇幅,在此不再赘述.

5 仿真结果

采用与文献[4]相同的产生规则,产生 n 个节点 ($n = 100 \sim 500$),弧的数量为 $3n$ 和 $6n$,规则如下:节点的坐标为二维均匀分布的整数,分布区间为 $[0, 120]$.为保持图的连通性,在图中产生一个生成树.这里将弧集 $\{e: (v_i, v_{i+1}), i = 1, 2, \dots, n-1\}$ 首先加入图中,其余的弧随机生成并添加到图中.弧的权重重定义为

$$C_{i,j} = [10 \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}]$$

其中 $[x]$ 表示大于等于 x 的最小的整数.节点的奖励 $p_j (j \in V)$ 为均匀分布在 $[1, 30]$ 中的整数.在每一个例子中,取相应的限额

$$Q = [\sum_{j \in V} p_j]$$

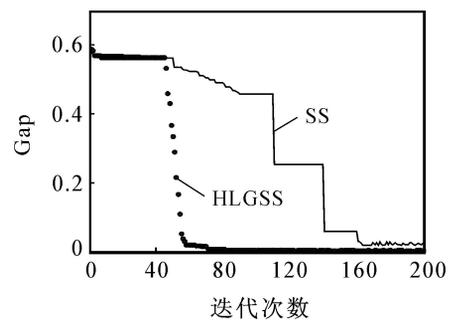


图 3 HL GSS 和 SS 收敛曲线($\alpha = 0.6, 500/5000$)

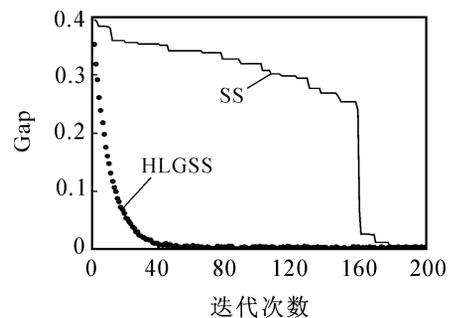


图 4 HL GSS 和 SS 收敛曲线($\alpha = 0.8, 500/5000$)

表 1 定额系数 = 0.6 的仿真结果

| 规模 $m \times n$ | 搜索空间减少比例(RP)/ % | Volume 算法 | | SS 算法 | | | HL GSS 算法 | | |
|-----------------|-----------------|-----------|--------|--------|--------|--------|-----------|--------|--------|
| | | LB | t/ s | Sol | t/ s | Gap/ % | Sol | t/ s | Gap/ % |
| 100/ 600 | 72 | 8 367 | 18.84 | 9 080 | 30.24 | 8.52 | 8 563 | 28.32 | 2.41 |
| 100/ 1 000 | 65 | 7 039 | 21.32 | 7 448 | 29.24 | 5.81 | 7 193 | 30.01 | 2.19 |
| 200/ 1 200 | 68 | 17 290 | 45.76 | 19 266 | 63.21 | 11.43 | 17 523 | 52.34 | 1.35 |
| 200/ 2 000 | 57 | 13 525 | 125.34 | 14 794 | 50.23 | 9.38 | 13 655 | 46.78 | 0.98 |
| 300/ 1 800 | 69 | 27 439 | 160.87 | 30 831 | 121.65 | 12.36 | 27 897 | 100.35 | 1.67 |
| 300/ 3 000 | 57 | 21 561 | 173.56 | 22 600 | 150.65 | 4.82 | 21 783 | 135.23 | 1.03 |
| 400/ 2 400 | 55 | 38 648 | 104.68 | 42 378 | 147.63 | 9.65 | 38 783 | 113.93 | 0.38 |
| 400/ 4 000 | 58 | 30 086 | 158.92 | 32 204 | 180.12 | 1.04 | 30 122 | 132.74 | 0.12 |
| 500/ 3 000 | 62 | 44 410 | 329.16 | 47 212 | 191.25 | 6.31 | 44 663 | 145.89 | 0.52 |
| 500/ 5 000 | 59 | 39 500 | 353.69 | 40 740 | 250.27 | 2.14 | 39 670 | 202.37 | 0.43 |
| 平均偏差 / % | | | | | 7.21 | | | 1.14 | |

表 2 定额系数 = 0.8 的仿真结果

| 规模 $m \times n$ | 搜索空间减少比例(RP)/ % | Volume 算法 | | SS 算法 | | | HL GSS 算法 | | |
|-----------------|-----------------|-----------|--------|--------|--------|--------|-----------|--------|--------|
| | | LB | t/ s | Sol | t/ s | Gap/ % | Sol | t/ s | Gap/ % |
| 100/ 600 | 64 | 10 329 | 20.85 | 10 870 | 35.23 | 5.24 | 10 468 | 30.21 | 1.37 |
| 100/ 1 000 | 63 | 9 385 | 24.03 | 9 686 | 42.58 | 3.21 | 9 448 | 40.13 | 0.65 |
| 200/ 1 200 | 71 | 29 810 | 54.48 | 32 240 | 103.25 | 8.15 | 29 926 | 87.54 | 0.39 |
| 200/ 2 000 | 69 | 19 890 | 85.52 | 20 511 | 120.32 | 3.12 | 19 914 | 100.58 | 0.12 |
| 300/ 1 800 | 54 | 33 875 | 101.13 | 35 057 | 180.56 | 3.49 | 33 885 | 153.82 | 0.03 |
| 300/ 3 000 | 59 | 28 370 | 108.48 | 28 733 | 231.39 | 1.28 | 28 523 | 176.35 | 0.55 |
| 400/ 2 400 | 31 | 51 530 | 230.12 | 53 174 | 160.12 | 3.19 | 51 607 | 130.32 | 0.18 |
| 400/ 4 000 | 58 | 42 374 | 157.43 | 42 764 | 280.32 | 0.92 | 42 556 | 204.35 | 0.43 |
| 500/ 3 000 | 42 | 68 323 | 345.27 | 69 013 | 304.27 | 1.01 | 68 507 | 224.78 | 0.29 |
| 500/ 5 000 | 47 | 54 861 | 305.32 | 55 207 | 360.54 | 1.63 | 54 872 | 243.24 | 0.02 |
| 平均偏差 / % | | | | | 3.11 | | | 0.41 | |

其中 α 是定额比例系数,这里分别取 $\alpha = 0.6, 0.8$, 这样共有 20 个仿真算例.

基于 Matlab 平台进行仿真,计算机为 Pentium IV 2.8 GHz, 512 M 内存. 仿真结果如表 1 和表 2 所示. 其中搜索空间减少比例(RP) = $(1 - |E| / |E|) * 100\%$, Gap = $(Sol - LB) / LB$, LB 为通过 Volume 算法得到的下界, t 为算法的运算时间.

从仿真结果可以看出(迭代次数均为 200 代), HL GSS 算法的所有偏差都保持在 3% 以内,而且随着定额系数 α 的增大,偏差逐渐减少,而 SS 算法的偏差则较大. 图 3 和图 4 为 HL GSS 和 SS 分别对于两个规模最大算例而进行的收敛性仿真分析. HL GSS 算法由于搜索空间缩小,在 40 ~ 60 代左右就已经收敛,而单独使用 SS 算法则在 140 ~ 180 代左右才收敛. 所以搜索空间的减少对于加速 SS 算法效果是

相当有利的.

6 结 语

实践证明,将启发式算法与数学规划相结合的混合算法是非常有效的. 本文提出的混合算法是对 SS 算法与数学规划相结合的一种尝试,本质上是对启发式算法和数学规划方法的浅层次组合. 未来的研究应着重进行两种甚至多种算法的有机组合,进而能够对算法的性能作出有界估计. 这类研究无论是在理论上还是应用上都具有十分重要的意义.

参考文献(References)

- [1] David S Johnson, Maria Minkoff, Steven Phillips. The prize collecting Steiner tree problem: Theory and practice[C]. Proc 11th ACM-SIAM Symp on Discrete Algorithm. San Francisco, 2000: 206-218.
- [2] Karp R M. Reducibility among combinatorial problems [M]. New York: Plenum Press, 1972: 85-103.

- [3] Abilio Lucena, Mauricio G C Resende. Strong lower bounds for the prize collecting Steiner problem in graphs [J]. *Discrete Applied Mathematics*, 2004, 141: 277-294.
- [4] Mohamed Haouari Jouhaina Chaouachi Siala. A hybrid Lagrangian genetic algorithm for the prize collecting Steiner tree problem [J]. *Computers & Operations Research*, 2006, 33: 1274-1288.
- [5] 王安顺, 郭子龙. 混沌免疫组合优化算法[J]. *控制与决策*, 2006, 21(2): 205-209.
(Wang An-shun, Guo Zi-long. Novel chaos immune optimization combination algorithm [J]. *Control and Decision*, 2006, 21(2): 205-209.)
- [6] Francisco Barahona, Ranga Anbil. The volume algorithm: Producing primal solutions with a subgradient method [J]. *Mathematical Programming*, 2000, (87): 385-399.
- [7] Bahiense L, Barahona F, Porto O. Solving Steiner tree problems in graphs with Lagrangian relaxation[J]. *J of Combinatorial Optimization*, 2003, 7: 259-282.
- [8] Dell 'Amico M, Maffioli F, Sciomachen A. A Lagrangian heuristic for the prize collecting travelling salesman problem[J]. *Annals of Operations Research*, 1998, 81: 289-305.
- [9] Raffensperger J F. Solving the TSP with decomposition-based pricing [C]. *Talk Presented at the 18th Int Symposium on Mathematical Programming*. Denmark: Copenhagen, 2003: 87-102.
- [10] Rafael Marti, Manuel Laguna, Fred Glover. Principles of scatter search [J]. *European J of Operational Research*, 2006, 169: 359-372.
- [11] Hamiez J P, Hao J K. Scatter search for graph coloring [J]. *Lecture Notes in Computer Science*, 2002, 23(10): 168-179.
- [12] 罗家祥, 唐立新. 带释放时间的并行机调度问题的 ILS & SS 算法 [J]. *自动化学报*, 2005, 31(6): 917-924.
(Luo Jia-xiang Tang Li-xin. A new ILS & SS algorithm for parallel-machine scheduling problem[J]. *Acta Automation Sinica*, 2005, 31(6): 917-924.)
- [13] Robert A Russell, Wen-Chyuan Chiang. Scatter search for the vehicle routing problem with time windows[J]. *European J of Operational Research*, 2006, 169: 606-622.
- [14] Christian Blum, Matthias Ehrgott. Local search algorithms for the k-cardinality tree problem [J]. *Discrete Applied Mathematics*, 2003, 128: 511-540.

(上接第 1340 页)

- [6] Arzi Y, Iaroslavitz L. Neural network-based adaptive production control system for a flexible manufacturing cell under a random environment[J]. *IIE Trans*, 1999, 31(3): 217-230.
- [7] 万国华. 用模糊调度系统求解动态 Job Shop 问题[J]. *系统工程理论与实践*, 2001, 21(8): 97-101.
(Wan Guo-hua. Using fuzzy scheduling system for solving a dynamic job shop problem [J]. *Systems Engineering Theory and Practice*, 2001, 21(8): 97-101.)
- [8] Piramuthu S, Shaw M, Fulkerson B. Information-base dynamic manufacturing system scheduling[J]. *Int J of Flexible Manufacturing Systems*, 2000, 12(2/3): 219-234.
- [9] Kim C O, Jun J, Baek J K, et al. Adaptive inventory control models for supply chain management[J]. *Int J of Advanced Manufacturing Technology*, 2005, 26(9/10): 1184-1192.
- [10] McDonnell P, Joshi S, Qiu R G. A learning approach to enhancing machine reconfiguration decision-making games in a heterarchical manufacturing environment [J]. *Int J of Production Research*, 2005, 43(20): 4321-4334.
- [11] Aydin M E, Öztemel E. Dynamic job-shop scheduling using reinforcement learning agents[J]. *Robotics and Autonomous Systems*, 2000, 33(2): 169-178.
- [12] Theodoridis S, Koutroumbas K. *Pattern recognition [M]*. 2nd Edition. San Diego: Academic Press, 2003.
- [13] Singh S, Jaakkola T, Littman M L, et al. Convergence results for single-step on-policy reinforcement-learning algorithms[J]. *Machine Learning*, 2000, 38(3): 287-308.