

具有全局性公平满意度的 最大多物资网络流问题^{*}

程从电

(沈阳师范大学数学与系统科学学院, 沈阳 110034)

(E-mail: zhiyang918@163.com)

李振鹏

(中国科学院数学与系统科学研究院系统科学研究所, 北京 100190)

(E-mail: lnlzhp2006@qq.com)

摘要 我们研究一个具有全局性公平满意度的最大多物资网络流问题 (MMFP-GFMR). 该项工作不仅丰富了最大多物资网络流问题的内容, 而且可用于研究某些实际优化决策问题, 例如运输过程中的一些资源分配问题. 文中主要内容如下: (A) 定义问题 MMFP-GFMR 并证明其解的存在性. (B) 设计一个求解 MMFP-GFMR 的拟多项式逼近算法. (C) 研究算法的复杂性与逼近程度. (D) 最后通过模拟计算验证了我们的工作.

关键词 网络; 多物资网络流; 资源分配; 最优决策; 逼近算法; 逼近程度

MR(2000) 主题分类 90C35; 90C27

中图分类 O221.7

1 引言

多物资网络流问题 (multicommodity flow problem), 简称多物资流问题, 是网络流 (network flow) 领域的一个重要分支, 其可用于处理诸多产生于紧急资源管理, 计算机科学技术, 通信和运输系统行为最优化, 生产分配计划, 网络设计以及金融流通等过程中的实际问题.

研究经典多物资流问题的变种或扩展性问题及其应用, 特别是在突发事件应急中的应用, 见 [1,2], 是该研究领域近年来的一项主要工作.

本文 2010 年 10 月 26 日收到. 2011 年 3 月 14 日收到修改稿.

* 辽宁省教育厅科研基金 (L2010514) 资助项目.

经典的最大物资流问题不考虑各种物资间的关系。可是,许多实际的最大物资流问题都强烈地涉及到各种物资之间的关系。一个典型的例子是,2008年初,中国政府紧急救援南方遭受特大雪灾地区行动中的救援物资分配问题。众所周知,在2008年中国人最重要的节日春节期间,中国许多通常不下雪的南部地区遭受了特大的雪灾。为了救援受灾地区,中国政府必须迅速地从北方的城市向南部的灾区运送大量的救援物资,例如,食品,棉衣和取暖设备等等;同时,还要紧急地疏散大量的被困人员回家过年。面对这种情况,有关决策者必须考虑下面的两个问题:一个是如何最大化救援物资的总量;另一个是如何平衡(或协调)不同地区之间的利益分配。换句话说,决策者需要做出一个既能够最大化救援物资的总量,又能够平衡各种需要的行动方案。我们如何才能够得到这样的一个最优决策呢?易知,从某种意义上讲,一个关于中国南方雪灾救援行动的方案可以被抽象地看作一个多物资流。所以,为了解决这一问题,我们可以先创建一个相应的多物资流问题,并设法求出其解,然后根据该解通过进一步的研究制定出救援的最优方案。抽象地讲,这一实际问题给出了一个需平衡各种物资需要的最大物资流问题。这里,由于技术的原因,我们不仅用“物资”一词表示货品,也用其表示点对发货点与收货点。此外,输油、送电、供煤和灌溉等过程也经常出现该问题。例如,2009年至2010年的冬季,世界上大部分地区由于气温异常偏低,取暖所耗能源大大超过往常同时期的用量;在这种境遇之下,合理地供应与分配能源,即需要平衡各种物资的流量的多物资流问题引起了广泛的关注。总而言之,连带考虑各种物资间某种关系的最大物资流问题值得认真地研究。类似的问题最近已有学者开始了探讨,见[3]。

2004年,中国著名数学家越民义先生在做关于网络流的理论与算法的学术报告时强调:在实践中,我们需要考虑各种多物资网络流问题,例如,如何求总流量最大且各种不同物资的流量比较均衡这样的多物资网络流问题;怎样寻找具有某种特殊性质的多物资网络流是一个很有前途的问题。上述事实表明,越先生的观点对于网络流研究的发展具有重要的指导意义。

一方面,需平衡各种物资流量的多物资流问题具有重要的实际意义;另一方面,由于“平衡各种物资的需要”这句话具有某种不确定性,就目前的形态而言,它只是一个描述性的问题,而不是一个严格的数学问题,所以在数学上是很难处理的。要想在理论上认真地研究这一问题,我们还需要建立一个能够表征该问题的严格的数学模型。

受到以上所述的启发,本文试给出和研究一个特殊的最大物资流问题,我们称其为具有全局性公平满意度的最大物资流问题(MMFP-GFMR)。该问题不仅在理论上丰富了最大物资流问题的内容,而且在实践上能够表征需要平衡各种物资流量的最大物资流问题,并可用于优化相关的实际问题。例如我们可以通过求解该问题来研究“中国南方雪灾救援”的最优行动方案。

本文剩余的内容组织如下:第2节给出必要的基础知识;第3节定义问题MMFP-GFMR,并证明其解存在;第4节设计求解MMFP-GFMR的逼近算法;第5节研究算法的复杂性与逼近程度;第6节汇报我们的计算试验;第7节扼要地总结全文。

2 预备知识

本节给出必要的基础知识.

设 $G = (V, E)$ 是一个图¹, 如果它的边既可以是有向的, 也可以是无向的, 我们就称其为一个混合图 (hybrid graph), 这里 V 和 E 分别表示 G 的所有的结点与所有的边. 我们将有向图和无向图都看做混合图的特例. 在本节与下一节里, 我们总假定相关的图为一个混合图.

给定一个具有权(或容量) c 的图 $G = (V, E)$ 和 $H = \{[s_i, t_i] : s_i, t_i \in V; i = 1, 2, \dots, k\}$, 这里 $c : E \rightarrow R_+$ 是权函数, $R_+ = [0, \infty)$, 而 s_i 和 t_i 分别表示物资 i 的发货点(源)与收货点(汇), 我们称三元组合 (G, c, H) 为一个多物资网络 (multicommodity network). 设 $[s_i, t_i] \in H$; $s_i, v_1, v_2, \dots, v_l, t_i$ 是 G 的一组除可能 $s_i = t_i$ 而外不能够彼此相等的结点; e_j 是 G 的分别以 v_j 和 v_{j+1} 为端点的边, $0 \leq j \leq l$, 这里 $v_0 = s_i$, $v_{l+1} = t_i$ (当 e_j 为有向边时, v_j 和 v_{j+1} 必须分别为 e_j 的始点与终点). 我们称 $P = [(s_i, e_0, v_1), (v_1, e_1, v_2), \dots, (v_l, e_l, t_i)]$ 为 (G, c, H) 的一条 $s_i - t_i$ -路径. (为了以后在记法上简单, 当 e_j 不需要指出时, 我们用 (v_j, v_{j+1}) 代表 (v_j, e_j, v_{j+1}) ; 当 v_j 和 v_{j+1} 不需要指出时, 我们用 e_j 代表 (v_j, e_j, v_{j+1})). 设 \mathcal{P}_i 是一族 $s_i - t_i$ -路径, 且对于所有的 $P \in \mathcal{P}_i$ 和 $e \in E(P)$ 都有 $c(e) > 0$, $i = 1, 2, \dots, k$, 这里 $E(P)$ 表示 P 的全体边. 我们称 $\mathcal{P} = \bigcup_{i=1}^k \mathcal{P}_i$ 为 (G, c, H) 上的一个(正向)的路径系统, 并用 $\mathcal{P}|_{(G, c, H)}$ 来表示, 或简记为 \mathcal{P} .

定义 1 设 \mathcal{P} 是一个路径系统. 如果映射 $y : \mathcal{P} \rightarrow R_+$ 满足:

$$\sum_{e \in E(P), P \in \mathcal{P}} y(P) \leq c(e), \quad \forall e \in E(\mathcal{P}),$$

这里 $E(\mathcal{P}) = \{e \in E(P) : P \in \mathcal{P}\}$, i.e. $E(\mathcal{P})$ 表示 \mathcal{P} 的所有边, 那么我们称 y 为 \mathcal{P} 上的一个多物资流, 简称为流(我们可以直观地将 $y(P)$ 理解为 y 沿着路经 P 的流量); 而称 $V(y) = \sum_{P \in \mathcal{P}} y(P)$ 为 y 的流量或流值, 称 $V_i(y) = \sum_{P \in \mathcal{P}_i} y(P)$ 为 y 的关于物资 i 的分流的流量或流值. 往后, 我们用 $F[\mathcal{P}]$ (或简单地用 F) 表示 \mathcal{P} 上的全体流.

注 1 众所周知, 关于(满足中介点流量守恒公式的)流有两种定义, 一种是用边集上的函数做定义; 另一种是像上面那样用路径系统上的函数做定义(这时, 所定义的流也满足中介点流量守恒公式, 见 [5]). 简单起见, 本文仅就由路径系统上函数所定义的流进行研究.

一般来说, 寻求满足某种条件的多物资流问题称为多物资流问题 (Multicommodity Flow Problem)(MFP). 下面是几个常见的多物资流问题. 问题在 \mathcal{P} 上寻找一个流 y 使得 $V(y)$ 达到最大, i.e. $V(y) = \hat{V} = \text{OPT}[\mathcal{P}] (= \max\{V(y) : y \in F[\mathcal{P}]\})$, 称为 \mathcal{P} 上的最大多物资流问题 (Maximum Multicommodity Flow Problem)(MMFP), 参见 [4]. 一个满足 MMFP

¹关于图的基础知识请参考 [4] 的第 2 章或其它有关资料.

的流 y 称为 MMFP 的解. 对于给定的 $b_i, i = 1, 2, \dots, k$, 寻找一个流 y 使得 $V_i(y) \leq b_i$ 且 $V(y)$ 达到最大, i.e. $V(y) = \bar{V}(\max\{V(y) : V_i(y) \leq b_i, i = 1, 2, \dots, k; y \in F\})$, 称为受限最大多物资流问题 (Maximum Flow Problem with Bounds)(MMFP-B). 对于给定的 $b_i, i = 1, 2, \dots, k$, 寻找一个流 y 使得 $V_i(y) = b_i$ 称为普通多物资流问题 (Common Flow Problem)(CMFP). 对于给定的 $b_i, i = 1, 2, \dots, k$, 寻找一个流 y 使得在 $V_i(y) = \lambda b_i, \lambda \in [0, 1]$ 的条件下 $V(y)$ 达到最大的问题, 称为最大一致多物资流问题 (Maximum Concurrent Flow Problem)(MCFP)^[6,7].

注 2 从某种意义上讲, 问题 MMFP-B, CMFP 和 MCFP 都可看做 MMFP 的衍生问题, 而 b_i 可以理解为关于物资 i 的需求量, 此外还可以将 (b_i) 理解为需求向量. 对于问题 MCFP 来说, 实际上 $\lambda = \frac{V_i(y)}{b_i}$, 我们可将其理解为流 y 关于物资 i 的满意率. 当 b_i 充分大以致 $b_i \geq \text{OPT}(\mathcal{P})$ 时, MMFP 和 MMFP-B 是等价的.

3 问题规划

现在我们开始明确地定义我们所要讨论的问题. 为了方便读者理解将要建立的模型, 我们先通过下面的例子对其意义做进一步的说明.

例 1 关于中国南方雪灾救援行动, 假定援助食品和御寒物资的城市分别为北京 (v_1) 和天津 (v_2), 而需要救援的地区分别为广州 (v_3), 上海 (v_4) 和武汉 (v_5). 再假定广州, 上海和武汉对于食品和御寒物资的需求量分别依次为 b_{13}, b_{14} 和 b_{15} 与 b_{23}, b_{24} 和 b_{25} ; 此外, 还假定这三座城市分别有 b_{31}, b_{41} 和 b_{51} 数量的被困人员需要向北京疏散. 在这种假定下, 考虑如何寻求该行动的最优决策. (简明起见, 这里我们假定救援是在一个确定的阶段上进行的. 此外, 我们还忽略了各种量的单位及其转换.)

分析: 设有关的交通系统形成一个图 G , 其各段路线的最大运输量形成该图的权函数 c . 用 $[s_{13}, t_{13}]$ 表示从 v_1 运送食品到 v_3 , $[s_{23}, t_{23}]$ 表示从 v_2 运送御寒物资到 v_3 , $[s_{31}, t_{31}]$ 表示从 v_3 疏散被困人员到 v_1 , 等等. 再令 $H = \{[s_{ij}, t_{ij}]\}$ 和 $\mathbf{b} = (b_{ij})$. 于是, $v_i, 1 \leq i \leq 5$ 都是 G 的节点, 而 (G, c, H) 构成一个多物资网络. 进而, 根据第一节的描述, 相关决策问题可粗略地转化为 (G, c, H) 上具有需求向量 \mathbf{b} 的需要平衡各种物资流量的最大多物资流问题.

由于 $\frac{V_{ij}(y)}{b_{ij}}$ 表示满意率, 为了解决该问题, 最好的办法当然是找出一个 MMFP-B 的解使得 $(V_{ij}(y)) = (\lambda b_{ij})$, 并且 $\lambda = \frac{\bar{V}}{\sum b_{ij}}$; 我们称这样的流为 MMFP-B 的理想解或一致点. 可是, 就通常情况来说, 由于受到要先保证流值最大的限制, 这样的解是找不到的; 而我们可以做到的只能是努力地调整 $\frac{V_{ij}(y)}{b_{ij}}$ 使得它们尽量的一致或均匀. 换句话说, 我们只能努力地寻找该问题的弱解或拟解. 为了实现平衡各种物资的需要这一目的, 一个实际的选择是从整体上使得 $(V_{ij}(y))$ 全局性地逼近理想解, 或说最小化 $\sum |V_{ij}(y) - \lambda b_{ij}|$. 也就是说, 通过最小化 $\sum |V_{ij}(y) - \lambda b_{ij}|$ 来表现平衡各种物资的需要.

显然, 一个满足上述条件的流能够强有力地帮助我们优化相关问题. 所以, 对于该救援行动来说, 我们可以先找出一个满足上述条件的流, 然后再根据它通过进一步的研究做出最优决策.

受第一节内容和本例的启发，现我们给出如下定义。

定义 2 设 \mathcal{P} 是多物资网络 (G, c, H) 上的一个路径系统，且 $\mathbf{b} = (b_1, b_2, \dots, b_k)$. 令 $\lambda = \bar{V}/\sum_{i=1}^k b_i$. 问题：在 \mathcal{P} 上寻找一个流 y 使得它是 MMFP-B 的一个解，并且 $\sum_{i=1}^k |V_i(y) - \lambda b_i|$ 最小，i.e. $V(y) = \bar{V}$ 且 $\sum_{i=1}^k |V_i(y) - \lambda b_i| = \min \left\{ \sum_{i=1}^k |V_i(y) - \lambda b_i| : V_i(y) \leq b_i; V(y) = \bar{V}, y \in F \right\}$, 称为具有全局性公平满意度的最大多物资流问题 (Maximum Multicommodity Flow Problem with Global Fair Met Rate)(MMFP-GFMR).

由例 1 可见，问题 MMFP-GFMR 的实际意义是明显的。此外该问题还具有重要的理论意义。首先，MMFP-GFMR 显然是一个特殊的 MMFP. 其次，MMFP-GFMR 和 MCFP 可以认为是由同一个实际问题所产生的。事实上，问题 CMFP 是一个既很实际又很理想的问题。易知，CMFP 的解既是 MMFP-B 的解，又是 MCFP 的解，这也就是说，CMFP 的解同时具有最大流和一致流的特征。但是，在一般情况下，它是没有解的。所以在实践中，我们经常倾向于最大性和一致性两者之一。当我们倾向于一致性时，我们将首先保证满意率一致，i.e. $V_i(y) = \lambda b_i$, 然后再促使流值尽量地大，这便产生了问题 MCFP. 而当我们倾向于最大性时，我们将首先保证流值最大，然后再促使满意率尽量地一致，这便粗略地产生了问题 MMFP-GFMR.

注 3 有时，由于理论与实际的差异以及实际的复杂性，理论研究只能为处理实际问题提供一个中介性的参考，我们认为这种研究将是今后数学发展的一个重要的方向。

注 4 当 $b_i \geq \text{OPT}(\mathcal{P})$, $1 \leq i \leq k$ 和 b_i , $1 \leq i \leq k$, 相等时，MCFP 和 MMFP-GFMR 分别为如下问题：寻求一个流 y 使得 $V(y) = \max\{V(y) : V_i(y) = V_1(y), i = 1, 2, \dots, k, y \in F\}$; 和寻求一个流 y 使得 $V(y) = \bar{V}$, $\sum_{i=1}^k |V_i(y) - \frac{1}{k}\bar{V}| = \min\{\sum_{i=1}^k |V_i(y) - \frac{1}{k}\bar{V}| : y \in F, V(y) = \bar{V}\}$. 后者可以理解为这样的一个问题：首先保证总的流值达到最大，然后再设法使得各个分流的流量尽量地算术平均。

定理 1 问题 MMFP-GFMR 的解存在。

证 易知 $B = \{y \in F : V(y) = \bar{V}, V_i(y) \leq b_i, i = 1, 2, \dots, k\} \neq \emptyset$. 当 B 的元素有限时，定理 1 的结论是平凡的。当 B 的元素无限时， $\inf \left\{ \sum_{i=1}^k |V_i(y) - \lambda b_i| : y \in B \right\}$ 存在并且有限，这里 $\lambda = \bar{V}/\sum_{i=1}^k b_i$. 设 $b = \inf \left\{ \sum_{i=1}^k |V_i(y) - \lambda b_i| : y \in B \right\}$, 则 B 中至少存在一个序列 $\{y_j\}$ 使得 $\lim_{j \rightarrow \infty} \left[\sum_{i=1}^k |V_i(y_j) - \lambda b_i| \right] = b$, 且对于所有的 $P \in \mathcal{P}$ 极限 $\lim_{j \rightarrow \infty} y_j(P)$ 都存在。再令 $y(P) = \lim_{j \rightarrow \infty} y_j(P)$, $\forall P \in \mathcal{P}$, 则 $y = (y(P)) \in B$, 并且 $\sum_{i=1}^k |V_i(y) - \lambda V| = b$. 这也就是说， y 是 MMFP-GFMR 的一个解。证毕。

4 算法

本节专门致力于设计一个求解问题 MMFP-GFMR 的逼近算法, 该算法是我们这项工作的最重要成果. 与该领域中以往的大部份工作的一个显著的区别是, 我们是通过建立辅助网络, 然后利用一个已知的逼近方案进行二分搜索的办法来设计我们的算法的. 为了简明, 从现在开始我们在有向图的框架下继续汇报我们的工作.

众所周知, 可用相应的 LP 来求解 MMFP, 但是, 一般来说, 由于相应的 LP 的规模太大, 这样做是很困难的, 所以寻找该问题的组合逼近算法是一项很令人兴奋的工作. 在过去的半个多世纪里, 这项工作一直是网络流领域里的一个主要的研究方向. 2000 年, 根据 [7] 的框架性工作, Korte 和 Vygen^[4] 给出了下面的算法 1 并证明了它是一个求 MMFP 的 ε - 逼近解 (ε - approximate solution, 见 [8]) 的完全多项式逼近方案 (full polynomial time approximation scheme)(FPTAS), 见 [4] 的 19.2.

算法 1 (关于 MMFP 的 FPTAS) 输入: 多物资网络 (G, c, H) , 路径系统 $\mathcal{P} (\neq \emptyset)$, 误差参数 ε ($0 < \varepsilon < \frac{1}{7}$). 输出: \mathcal{P} 上的具有定理 2(见下节) 所述性质的流 y^* .

1. 置 $y(P) = 0, \forall P \in \mathcal{P}$. 置 $z(e) = \delta = [n(1 + \varepsilon)]^{-[\frac{5}{\varepsilon}]}(1 + \varepsilon), \forall e \in E(\mathcal{P})$.
2. 置 $z[P] := \sum_{e \in E(P)} z(e), \forall P \in \mathcal{P}$. 然后在 \mathcal{P} 中寻找 P^* 使得 $z[P^*] = \min\{z[P] : P \in \mathcal{P}\}$. 当 $z[P^*] \geq 1$ 时, 转 4.
3. 置 $\gamma := \min\{c(e) : e \in E(P^*)\}, y(P^*) := y(P^*) + \gamma$, 且 $z(e) := z(e)[1 + \frac{\varepsilon\gamma}{c(e)}], \forall e \in E(P^*)$. 转 2.
4. 令 $\xi = \max\{\frac{1}{c(e)} \sum_{e \in E(P), P \in \mathcal{P}} y(P) : e \in E(\mathcal{P})\}$, 且 $y^*(P) = \frac{y(P)}{\xi}, \forall P \in \mathcal{P}$.

注 5 关于算法 1, 根据第五节的定理 2, $V(y^*) \geq \frac{1}{1+\varepsilon} \hat{V} = (1 - \frac{\varepsilon}{1+\varepsilon}) \hat{V}$. 这表明 y^* 是一个 $(\frac{\varepsilon}{1+\varepsilon})$ - 逼近解. 由于 ε 是任意的, 算法 1 实际上是一个 ε - 逼近算法.

根据算法 1, 按照下述的方法, 我们可以给出一个关于 MMFP-GFMR 的逼近方案. 首先, 通过构造辅助网络, 利用算法 1, 建立一个关于 MMFP-B(和 CMFP) 的 ε - 逼近算法, 即如下算法 2. 然后, 依次通过由算法 2 求出必要的中间变量, 构造辅助网络, 和利用算法 1 反复地进行二分搜索, 设计出关于 MMFP-GFMR 的一个逼近方案, 即如下的算法 3.

算法 2 (关于 MMFP-B (CMFP) 的 FPTAS) 输入: 多物资网络 (G, c, H) , 路径系统 \mathcal{P} , 向量 \mathbf{b} 和误差参数 ε ($0 < \varepsilon < \frac{1}{7}$). 输出: \mathcal{P} 上的具有定理 3 所述性质的流 y^* .

1. 置 $\tilde{V} = V \cup \{\tilde{t}_i : i = 1, 2, \dots, k\} (\tilde{t}_i \in V)$,

$$\begin{aligned} \tilde{E} &= E \cup \{(t_i, \tilde{t}_i) : i = 1, 2, \dots, k\}, \\ \tilde{H} &= \{[s_i, \tilde{t}_i] : i = 1, 2, \dots, k\}, \end{aligned}$$

$$\begin{aligned}\tilde{c}(e) &= \begin{cases} c(e), & e \in E, \\ b_i, & e = (t_i, \tilde{t}_i), \quad i = 1, 2, \dots, k, \end{cases}, \quad \tilde{G} = (\tilde{V}, \tilde{c}, \tilde{E}), \\ \tilde{\mathcal{P}}_i &:= \{[(s_i, v_1), (v_1, v_2), \dots, (v_l, t_i), (t_i, \tilde{t}_i)] \\ &= P + (t_i, \tilde{t}_i) : P = [(s_i, v_1), (v_1, v_2), \dots, (v_l, t_i)] \in \mathcal{P}_i\}, \quad i = 1, 2, \dots, k; \\ \tilde{\mathcal{P}} &:= \bigcup_{i=1}^k \tilde{\mathcal{P}}_i.\end{aligned}$$

2. 关于 $\tilde{\mathcal{P}}|_{(\tilde{G}, \tilde{c}, \tilde{H})}$ 上的 MMFP 与输入 ε , 用算法 1 求出 \tilde{y}^* .

3. 置 $y^* = \tilde{y}^*(P + (t_i, \tilde{t}_i))$, $\forall P \in \mathcal{P}_i$, $i = 1, 2, \dots, k$.

注 6 这里, 我们通过一个简单的例子来初略地说明一下算法 2 的思想.

关于 G 为图 1 的网络 (G, c, H) , 给定 $\mathbf{b} = (b_1, b_2)$, 令

$$\begin{aligned}\mathcal{P}_1 &= \{[(s_1, t_1)], [(s_1, s_2), (s_2, t_1)]\}, \\ \mathcal{P}_2 &= \{[(s_2, t_1), (t_1, t_2)], [(s_2, s_1), (s_1, t_2)]\}, \quad \mathcal{P} = \mathcal{P}_1 \cup \mathcal{P}_2.\end{aligned}$$

考虑 MMFP-B.

作 \tilde{G} (图 2), 并通过定义 \tilde{c} 建构辅助网络 $(\tilde{G}, \tilde{c}, \tilde{H})$, 这里

$$\tilde{c}(e) = \begin{cases} c(e), & e \neq (t_i, \tilde{t}_i), \quad i = 1, 2; \\ b_i, & e = (t_i, \tilde{t}_i), \quad i = 1, 2. \end{cases}$$

再定义 $\tilde{\mathcal{P}}$ 如下:

$$\begin{aligned}\tilde{\mathcal{P}}_1 &= \{[(s_1, t_1), (t_1, \tilde{t}_1)], [(s_1, s_2), (s_2, t_1), (t_1, \tilde{t}_1)]\}, \\ \tilde{\mathcal{P}}_2 &= \{[(s_2, t_1), (t_1, t_2), (t_2, \tilde{t}_2)], [(s_2, s_1), (s_1, t_2), (t_2, \tilde{t}_2)]\}.\end{aligned}$$

根据 \mathcal{P} 与 $\tilde{\mathcal{P}}$ 的关系, 设 \tilde{y} 是 MMFP($\tilde{\mathcal{P}}$)(关于 $\tilde{\mathcal{P}}$ 的 MMFP) 的解, 并令 $y(P) = \tilde{y}(P + (t_i, \tilde{t}_i))$, $\forall P \in \mathcal{P}_i$, $i = 1, 2$, 这里 $(P + (t_i, \tilde{t}_i))$ 表示 $[\dots, (t_i, \tilde{t}_i)]$, 而 $P = [\dots]$, 则 y 是 MMFP-B 的解. 因此, 我们可以通过用算法 1 求 MMFP($\tilde{\mathcal{P}}$) 的逼近解来求 MMFP-B 的逼近解. 这就是我们设计算法 1 的思想.

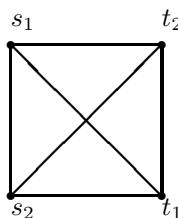


图 1

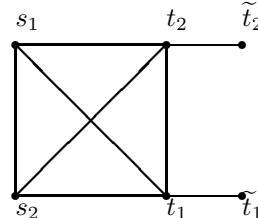


图 2

算法 3 (关于 MMFP-GFMR 的逼近算法) 输入: 多物资网络 (G, c, H) , 路径系统 \mathcal{P} , 向量 \mathbf{b} 和误差参数 $\eta > 0$. 输出: \mathcal{P} 上的具有定理 4 所述性质的流 y^* .

1. 取 $\varepsilon_0 \in (0, \frac{1}{7})$. 关于 \mathcal{P} , \mathbf{b} 和 ε_0 , 用算法 2 求出 MMFP-B 的一个逼近解 y_0^* . 如果 $(1 + \varepsilon_0)V(y_0^*) \leq \eta$, 置 $y^* = y_0^*$. 停机. 否则, 置 $\varepsilon = \min\{\frac{\eta}{(1 + \varepsilon_0)V(y_0^*)}, \varepsilon_0\}$. 然后, 关于 ε 再次运用算法 2 求出 MMFP-B 的一个逼近解 y_1^* . 最后, 置 $U = V(y_1^*) + \eta$ (注意 $U \geq \bar{V}$, 请参考定理 4 的证明.)
2. 置 $\tilde{V} = V \cup \{t_0, \tilde{t}_0; t'_i, \tilde{t}_i : i = 1, 2, \dots, k\}$ ($t_0, \tilde{t}_0, t'_i, \tilde{t}_i \in V$),

$$\begin{aligned}\tilde{E} &= E \bigcup \{(t_i, t'_i), (t'_i, \tilde{t}_i), (t'_i, t_0), (t_0, \tilde{t}_0) : i = 1, 2, \dots, k\}, \\ \tilde{H} &= \{[s_i, \tilde{t}_i], [s_i, \tilde{t}_0] : i = 1, 2, \dots, k\}, \\ \tilde{G} &= (\tilde{V}, \tilde{E}); \quad \tilde{\mathcal{P}}_i = \{P + (t_i, t'_i) + (t'_i, \tilde{t}_i) : P \in \mathcal{P}_i\}, \\ \tilde{\mathcal{P}}_0^i &= \{P + (t_i, t'_i) + (t'_i, t_0) + (t_0, \tilde{t}_0) : P \in \mathcal{P}_i\}, \quad i = 1, 2, \dots, k; \\ \tilde{\mathcal{P}} &= \left(\bigcup_{i=1}^k \tilde{\mathcal{P}}_i \right) \bigcup \left(\bigcup_{i=1}^k \tilde{\mathcal{P}}_0^i \right); \\ \tilde{\lambda} &= \frac{U}{\sum b_i}, \quad b := 0, \\ \tilde{c}(e) &= \begin{cases} c(e), & e \in E(G), \\ b_i, & e = (t_i, t'_i), \\ \tilde{\lambda}b_i, & e = (t'_i, \tilde{t}_i) \\ U, & e = (t'_i, t_0), \quad i = 1, 2, \dots, k, \\ b, & e = (t_0, \tilde{t}_0). \end{cases}\end{aligned}$$

3. 关于 $\tilde{\mathcal{P}}$ 和 ε , 用算法 1 求出 MMFP 的一个逼近解 \tilde{y}^* . (注意: 这时, 由于 $b = 0$, $(\bigcup_{i=1}^k \tilde{\mathcal{P}}_0^i)$ 是无效的, 事实上 $\tilde{\mathcal{P}} = (\bigcup \tilde{\mathcal{P}}_i)$.) 如果 $V(\tilde{y}^*) \geq \frac{1}{1+\varepsilon}V(y_1^*)$, 置 $\tilde{y}_1^* := \tilde{y}^*$. 然后转 6. 否则, 置 $\tilde{y}_2^* := \tilde{y}^*, a := 0, b := U$. 关于 $\tilde{\mathcal{P}}$ 和 ε , 再用算法 1 求出 MMFP 的一个逼近解 \tilde{y}_1^* . 然后进行下一步. (注意: 由于 $U \geq \bar{V}$, $V(\tilde{y}_1^*) \geq \frac{1}{1+\varepsilon}\bar{V} \geq \frac{1}{1+\varepsilon}V(y_1^*)$.)
4. 置 $b := \frac{a+b}{2}$.
5. 关于 $\tilde{\mathcal{P}}$ 和 ε , 用算法 1 求出 MMFP 的一个逼近解 \tilde{y}^* .
 - (i) $V(\tilde{y}^*) \geq \frac{1}{1+\varepsilon}V(y_1^*)$, 置 $\tilde{y}_1^* := \tilde{y}^*$. 如果 $(b - a) < \eta$, 转 6. 否则, 转 4.
 - (ii) $V(\tilde{y}^*) < \frac{1}{1+\varepsilon}V(y_1^*)$, 置 $\tilde{y}_2^* := \tilde{y}^*, a' := b, b := 2b - a, a := a'$. 如果 $(b - a) < \eta$, 转 6. 否则, 转 4.
6. 置 $y^*(P) = \tilde{y}_1^*(P + (t_i, t'_i) + (t'_i, \tilde{t}_i)) + \tilde{y}_1^*(P + (t_i, t'_i) + (t'_i, t_0) + (t_0, \tilde{t}_0)), \forall P \in \mathcal{P}_i, i = 1, 2, \dots, k$. 停机.

注 7 关于注 6 中的 (G, c, H) , $\mathbf{b} = (b_1, b_2)$ 和 \mathcal{P} , 考虑如何求 MMFP-GFMR 的逼近解.

首先做 \tilde{G} (图 3), 并通过定义 \tilde{c} 与 \tilde{H} 建构辅助网络 $(\tilde{G}, \tilde{c}, \tilde{H})$, 这里, $\tilde{c}((t_1, t'_1)) = b_1, \tilde{c}((t'_1, \tilde{t}_1)) = \lambda b_1, \tilde{c}((t_2, t'_2)) = b_2, \tilde{c}((t'_2, \tilde{t}_2)) = \lambda b_2, \tilde{c}((t'_1, t_0)) = \tilde{c}((t'_2, t_0)) = \bar{V}, \tilde{c}((t_0, \tilde{t}_0)) = b$ (变量), 和 $\tilde{c}(e) = c(e) \quad \forall e \in E(G); \tilde{H} = \{[s_1, \tilde{t}_1], [s_1, \tilde{t}_0], [s_2, \tilde{t}_2], [s_2, \tilde{t}_0]\}$. 再定义 $\tilde{\mathcal{P}}$ 如下, $\tilde{\mathcal{P}} = \tilde{\mathcal{P}}_1 \cup \tilde{\mathcal{P}}_1^0 \cup \tilde{\mathcal{P}}_2 \cup \tilde{\mathcal{P}}_2^0$, 且 $\tilde{\mathcal{P}}_1 = \{P + (t_1, t'_1) + (t'_1, \tilde{t}_1) : P \in \mathcal{P}_1\}, \tilde{\mathcal{P}}_1^0 = \{P + (t_1, t'_1) + (t'_1, t_0) + (t_0, \tilde{t}_0) : P \in \mathcal{P}_1\}, \tilde{\mathcal{P}}_2 = \{P + (t_2, t'_2) + (t'_2, \tilde{t}_2) : P \in \mathcal{P}_2\}, \tilde{\mathcal{P}}_2^0 = \{P + (t_2, t'_2) + (t'_2, t_0) + (t_0, \tilde{t}_0) : P \in \mathcal{P}_2\}$.

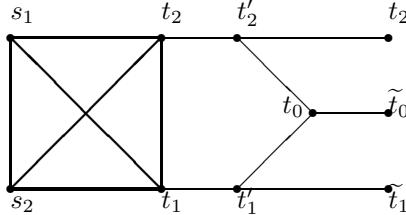


图 3

设 y 是 MMFP-B 的一个解, 则 $V(y) = V_1(y) + V_2(y) = \bar{V} = \lambda(b_1 + b_2)$, 这里 $\lambda = \frac{\bar{V}}{b_1 + b_2}$. 因此, $(V_1(y) - \lambda b_1) = -(V_2(y) - \lambda b_2)$. 设 $(V_1(y) - \lambda b_1) > 0$, 则 $\sum |V_i(y) - \lambda b_i| = 2(V_1(y) - \lambda b_1) = 2[\max\{V_1(y) - \lambda b_1, 0\} + \max\{V_2(y) - \lambda b_2, 0\}]$. 从而, y 为 MMFP-GFMR 的解充分必要 $[\max\{V_1(y) - \lambda b_1, 0\} + \max\{V_2(y) - \lambda b_2, 0\}]$ 最小. 注意到 \tilde{t}_0 可以看做一个储存进入 (t_1, t'_1) 而不能够进入 (t'_1, \tilde{t}_1) (或进入 (t_2, t'_2) 而不能够进入 (t'_2, \tilde{t}_2)) 的流的库房. 令 $w = \frac{1}{2} \min\{\sum |V_i(y) - \lambda b_i| : V_i(y) \leq b_i, y \in F[\mathcal{P}]\}$, 则:

(i) 当 $b < w$ 时, $\text{OPT}[\tilde{\mathcal{P}}] \leq \bar{V}$ ($\bar{V} = \max\{V(y) : V_i(y) \leq b_i, i = 1, 2; y \in F[\mathcal{P}]\}$). 因此, 关于 $\tilde{\mathcal{P}}$ 和充分小的 ε , 用算法 1 所求出的流 \tilde{y} 应不满足 $V(\tilde{y}) \geq \frac{1}{1+\varepsilon}\bar{V}$.

(ii) 当 $b = w$ 时, 设 \tilde{y} 是 MMFP($\tilde{\mathcal{P}}$) 的解, 并命

$$y(P) = \begin{cases} \tilde{y}(P + (t_1, t'_1) + (t'_1, \tilde{t}_1)) \\ + \tilde{y}(P + (t_1, t'_1) + (t'_1, t_0) + (t_0, \tilde{t}_0)), & P \in \mathcal{P}_1; \\ \tilde{y}(P + (t_2, t'_2) + (t'_2, \tilde{t}_2)) \\ + \tilde{y}(P + (t_2, t'_2) + (t'_2, t_0) + (t_0, \tilde{t}_0)), & P \in \mathcal{P}_2, \end{cases}$$

则 y 是 MMFP-GFMR 的解.

(iii) 当 $b > w$ 时, $\text{OPT}[\tilde{\mathcal{P}}] = \bar{V}$. 因此, 关于 $\tilde{\mathcal{P}}$ 和充分小的 ε , 用算法 1 所求出的流 \tilde{y} 应满足 $V(\tilde{y}) \geq \frac{1}{1+\varepsilon}\bar{V}$.

根据 (ii) 与算法 1, 只要找到了 w 就可以求得 MMFP-GFMR 的逼近解. 根据 (i) 与 (iii), 取充分小的 ε 和区间 (b_1, b_2) 使得当 $b = b_1$ 时用算法 1 所求出的流 \tilde{y} 不满足 $V(\tilde{y}) \geq \frac{1}{1+\varepsilon}\bar{V}$; 而当 $b = b_2$ 时用算法 1 所求出的流 \tilde{y} 满足 $V(\tilde{y}) \geq \frac{1}{1+\varepsilon}\bar{V}$, 则 $w \in (b_1, b_2)$. 从而, 可以通过逐渐地缩短 (b_1, b_2) 的长度近似地找到 w . 算法 3 就是遵照这种思想而设计出来的.

算法 3 之关键在于建立辅助网络和利用算法 1 循环地进行二分搜索. 这种方法是它不同于以往的相关算法的主要特征. 值得注意的是, 该方法可用于求解其它的网络流问题.

5 算法分析

本节讨论上节所给出的算法的正确性, 逼近度及复杂性.

定理 2 (Theorem 19.6 of [4]) 算法 1 的复杂性为: $O(\frac{1}{\varepsilon^2} km^2 \log n)$, 这里 $k = |H|$, $n = |V(G)|$, $m = |E(G)|$. 对于一个给定的问题 MMFPF 和 $\varepsilon \in (0, \frac{1}{7})$, 设 y^* 为算法 1 的输出, 则 y^* 是 \mathcal{P} 上的一个流, 且

$$V(y^*) \geq \frac{1}{1+\varepsilon} \text{OPT}(\mathcal{P})(= \widehat{V}). \quad (1)$$

证 请参见 [4] 的 Theorem 19.6 及其证明, 或 [5]. 证毕.

注 8 算法 1 和定理 2 分别是 [4] 的第十九章中的 Multicommodity Flow Approximation Scheme 和 Theorem 19.6. 本文通过如下变动对它们进行了改进. (i) 在 [4] 的这两项内容中, $\varepsilon \in (0, \frac{1}{2}]$; 而在本文的算法 1 和定理 2 中, $\varepsilon \in (0, \frac{1}{7}]$. (ii) 在 [4] 的 Theorem 19.6 中, 算法的运行时间为: $O(\frac{1}{\varepsilon^2} kmn^2 \log n)$; 而在本文的定理 2 中, 算法的运行时间为: $O(\frac{1}{\varepsilon^2} km^2 \log n)$. 关于我们所做的变动的理由与正确性请参见 [5].

如上所述, 算法 1 的输出 y^* 是给定的 MMFP 的一个 ε -逼近解. 由于对于给定的 ε 来说, $O(\frac{1}{\varepsilon^2} km^2 \log n) = O(km^2 \log n)$, 定理 2 表明算法 1 是一个求 MMFP 的 ε -逼近解的 FPTAS.

定理 3 算法 2 的复杂性与算法 1 的复杂性相同. 设 y^* 是算法 2 的输出, 则 y^* 是 \mathcal{P} 上的一个流, 且 $V(y^*) \geq \frac{1}{1+\varepsilon} \overline{V}$, $V_i(y^*) \leq b_i$, $i = 1, 2, \dots, k$ (即 y^* 是关于输入 \mathcal{P} 和 \mathbf{b} 的 MMFP-B 的一个 ε -逼近解. 另外, 如果相应的 CMFP 的解存在, 则 $V(y^*) \geq \frac{1}{1+\varepsilon} (\sum b_i)$, 这也就是说, y^* 是相应的 CMFP 的一个 ε -逼近解.)

证 见 [5]. 证毕.

定理 4 算法 3 的复杂性为 $O((\frac{1}{\varepsilon^2} km^2 \log n) \cdot (\log \frac{\overline{V}}{\eta}))$, 这里 $\varepsilon = \min\{\varepsilon_0, \eta[(1+\varepsilon_0)\overline{V}]^{-1}\}$. 设 y 为相应的问题 MMFP-GFMR 的解, y^* 为算法 3 的输出, 且

$$Y_i = \max\{V_i(y) - \lambda b_i, 0\}, \quad Y_i^* = \max\{V_i(y^*) - \lambda b_i, 0\}, \quad i = 1, 2, \dots, k.$$

这里 $\lambda = \frac{\overline{V}}{\sum b_i}$, 则 y^* 为 \mathcal{P} 上的一个流, 并且

$$V(y) \geq V(y^*) \geq V(y) - 2\eta; \quad \sum Y_i^* \leq \sum Y_i + 2\eta. \quad (2)$$

注意: $\sum Y_i = \frac{1}{2} \sum |V_i(y) - \lambda b_i|$. 事实上,

$$\begin{aligned} \sum V_i(y) &= \overline{V} = \lambda \left(\sum b_i \right) \Rightarrow \sum_{V_i(y) \geq \lambda b_i} [V_i(y) - \lambda b_i] \\ &= \sum_{V_i(y) < \lambda b_i} [\lambda b_i - V_i(y)] \Rightarrow \sum Y_i = \frac{1}{2} \sum |V_i(y) - \lambda b_i|. \end{aligned}$$

证 显然, 算法 3 的复杂性取决于它调用其子程序的次数, 及其子程序的复杂性. 易知, 算法 3 调用其子程序的次数为 $O(\log \frac{\overline{V}}{\eta})$ (or $O(\log \frac{\max b_i}{\eta})$). 另一方面, 根据定理

3, 其子程序算法 1 和算法 2 的复杂性为 $O(\frac{1}{\varepsilon^2} km^2 \log n)$. 所以, 算法 3 的复杂性为: $O((\frac{1}{\varepsilon^2} km^2 \log n) \cdot (\log \frac{\bar{V}}{\eta}))$.

根据定理 2, \tilde{y}^* 是 $\tilde{\mathcal{P}}$ 上的一个流. 由此, 根据 y^* 与 \tilde{y}^* 的关系可知, y^* 是 \mathcal{P} 上的一个流.

由于 y 是 MMFP-GFMR 的一个解, 我们有 $V(y) = \bar{V} \geq V(y_1^*)$. 根据定理 3, $V(y_0^*) \geq \frac{1}{1+\varepsilon_0} \bar{V}$ 和 $V(y_1^*) \geq \frac{1}{1+\varepsilon} \bar{V}$. 因此, $\bar{V} - V(y_1^*) \leq \bar{V} - \frac{1}{1+\varepsilon} \bar{V} = \frac{\varepsilon}{1+\varepsilon} \bar{V} \leq \varepsilon \bar{V} \leq \frac{\eta \bar{V}}{(1+\varepsilon_0)V(y_0^*)} \leq \eta$, 进而 $V(y_1^*) \leq \bar{V} \leq V(y_1^*) + \eta = U$ (关于 \bar{V} 的意义, 见 MMFP-B; 关于 $\varepsilon, V(y_1^*)$ 和 U 的意义, 见算法 3 的步 1).

1. 运算在步 1 停止. 这时, $y^* = y_0^*$. 根据定理 3, $V(y_0^*) \geq \frac{1}{1+\varepsilon_0} \bar{V}$. 因此, $\bar{V} \leq (1+\varepsilon_0)V(y_0^*)$. 由于 $(1+\varepsilon_0)V(y_0^*) \leq \eta$, 我们进一步有 $\bar{V} \leq \eta$. 于是, $\bar{V} \geq V(y^*) \geq \bar{V} - 2\eta$, 且 $\sum Y_i^* \leq V(y^*) \leq \eta \leq \sum Y_i + 2\eta$. 即 (2) 成立.

2. 运算经步 3 转步 6 后停止. 这时, 由于 $b = 0$, $(\bigcup_{i=1}^k \tilde{\mathcal{P}}_0^i)$ 是无效的, 所以 $V(y^*) = V(\tilde{y}^*) \geq \frac{1}{1+\varepsilon} V(y_1^*)$. 因此,

$$\begin{aligned} \bar{V} \geq V(y^*) &\geq \frac{1}{1+\varepsilon} V(y_1^*) \geq \frac{1}{1+\varepsilon} \cdot \frac{1}{1+\varepsilon} \bar{V} = \bar{V} - \left(\bar{V} - \frac{1}{1+\varepsilon} \cdot \frac{1}{1+\varepsilon} \bar{V} \right) \\ &= \bar{V} - \bar{V} \left(\frac{\varepsilon}{1+\varepsilon} \cdot \frac{2+\varepsilon}{1+\varepsilon} \right) \geq \bar{V} - \bar{V} \cdot 2\varepsilon \geq \bar{V} - \bar{V} \cdot \frac{2\eta}{(1+\varepsilon_0)V(y_0^*)} \\ &\geq \bar{V} - 2\eta \quad \left(\text{注意: } V(y_0^*) \geq \frac{1}{(1+\varepsilon_0)} \bar{V} \right). \end{aligned} \tag{3}$$

另一方面,

$$\begin{aligned} \sum Y_i^* &= \sum \max\{V_i(y^*) - \lambda b_i, 0\} = \sum \max\{V_i(\tilde{y}^*) - \lambda b_i, 0\} \\ &\leq \sum (\tilde{\lambda} b_i - \lambda b_i) = U - \bar{V} = [V(y_1^*) + \eta] - \bar{V} \leq \eta < \sum Y_i + 2\eta. \end{aligned}$$

所以, (2) 成立 (关于 $\tilde{\lambda}$ 的定义, 见步 2).

3. 运算经步 5 转步 6 后停止. 首先, 根据 $\bar{V} \leq U$ 和 $(\tilde{G}, \tilde{c}, \tilde{H})$ 的建立方法, 对于步 3 与最终的 $\tilde{y}_1^*, \tilde{y}_2^*$ 和 b , 我们能够容易地得到

$$\begin{aligned} V(\tilde{y}_1^*) &\geq \frac{1}{1+\varepsilon} V(y_1^*), \quad V(\tilde{y}_2^*) < \frac{1}{1+\varepsilon} V(y_1^*), \\ \sum_{i=1}^k \sum_{P \in \mathcal{P}_i} \tilde{y}_1^*(P + (t_i, t'_i) + (t'_i, t_0) + (t_0, \tilde{t}_0)) &\leq b. \end{aligned} \tag{4}$$

根据 (4), 我们可以进一步地知道 $\sum Y_i > a$ (这里, a 是最终的). 事实上, 如果 $\sum Y_i \leq a$, 那么根据 y 是 MMFP-GFMR 的解, $V(y) \leq U$ 和 $(\tilde{G}, \tilde{c}, \tilde{H})$ 的建立方法, 我们可以知道: 当 $\tilde{c}((t_0, \tilde{t}_0)) = a$ 时, 有 $\text{OPT}(\tilde{\mathcal{P}}) = V(y) = \bar{V}$. 这也就是说, 当 $\tilde{c}((t_0, \tilde{t}_0)) = a$ 时, 对于 $\tilde{\mathcal{P}}$ 上的 MMFP, 用算法 1 所求出的解 \tilde{y}^* 满足: $V(\tilde{y}^*) \geq \frac{1}{1+\varepsilon} \bar{V} \geq \frac{1}{1+\varepsilon} V(y_1^*)$. 由此可知, $V(\tilde{y}_2^*) \geq \frac{1}{1+\varepsilon} V(y_1^*)$. 这与 $V(\tilde{y}_2^*) < \frac{1}{1+\varepsilon} V(y_1^*)$ 矛盾. 所以, $\sum Y_i > a$.

回顾 y^* 与 \tilde{y}_1^* 的关系, 我们得到

$$\begin{aligned}
 \sum Y_i^* &\leq \sum_{i=1}^k \sum_{P \in \mathcal{P}_i} \tilde{y}_1^*(P + (t_i, t'_i) + (t'_i, t_0) + (t_0, \tilde{t}_0)) \\
 &\quad + \sum_{i=1}^k \max \left\{ \left[\sum_{P \in \mathcal{P}_i} \tilde{y}_1^*(P + (t_i, t'_i) + (t'_i, \tilde{t}_i)) \right] - \lambda b_i, 0 \right\} \\
 &\leq b + \sum_{i=1}^k (\tilde{\lambda} b_i - \lambda b_i) = (b - a) + a + \sum_{i=1}^k (\tilde{\lambda} b_i - \lambda b_i) \\
 &\leq \eta + \sum Y_i + (U - \bar{V}) \leq \sum Y_i + \eta + [(V(y_1^*) + \eta) - \bar{V}] \\
 &\leq \sum Y_i + 2\eta, \quad \lambda = \frac{\bar{V}}{\sum b_i}, \quad \tilde{\lambda} = \frac{U}{\sum b_i}
 \end{aligned}$$

(\tilde{y}_1^* 和 b 是最终的).

注意:

$$\begin{aligned}
 \sum_{i=1}^k \sum_{P \in \mathcal{P}_i} \tilde{y}_1^*(P + (t_i, t'_i) + (t'_i, t_0) + (t_0, \tilde{t}_0)) &\leq b, \\
 \sum_{i=1}^k \max \left\{ \left[\sum_{P \in \mathcal{P}_i} \tilde{y}_1^*(P + (t_i, t'_i) + (t'_i, \tilde{t}_i)) \right] - \lambda b_i, 0 \right\} &\leq \sum_{i=1}^k (\tilde{\lambda} b_i - \lambda b_i)
 \end{aligned}$$

且 $b - a < \eta$.

此外, 根据

$$V(y) = \bar{V} \geq V(y^*) = V(\tilde{y}_1^*) \geq \frac{1}{1+\varepsilon} V(y_1^*),$$

由 (3) 可知 $V(y) \geq V(y^*) \geq V(y) - 2\eta$. 综上, (2) 成立.

最后, 注意到算法 3 不会在其它的情况停机, 根据以上过程便知, 定理 4 成立. 证毕.

因为在数量级别上 $O(\frac{1}{\varepsilon^2} km^2 \log n)$ 与 $O((\frac{1}{\varepsilon^2} km^2 \log n) \cdot (\log \frac{\bar{V}}{\eta}))$ 是没有差别的, 算法 3 与算法 1 实际上是等效的. 另外, 由于图 G 中可能有平行边, 不宜单用 n 刻划输入规模, 所以我们用 n 和 m 来共同表示输入的规模. 又由于 $O((\frac{1}{\varepsilon^2} km^2 \log n) \cdot (\log \frac{\bar{V}}{\eta}))$ 中的 \bar{V} 与其它的输入有关, 所以算法 3 只是一个求解 MMFP-GFMR 的拟多项式逼近算法 (关于拟多项式算法的定义可参考 [9] 第 210 页的有关叙述).

6 计算实验

为了验证算法 3 及定理 4, 本节演示两个运用算法 3 进行计算的实际例子. 通过这两个实例我们还进一步解释了问题 MMFP-GFMR. 我们的计算是通过在一台 Intel(R) Pentium(R) Dual E2160 @ 1.80GHz 1.80 GHz, 0.99GB RAM 的个人计算机上运行算法 3 的 Matlab 语言程序而完成的.

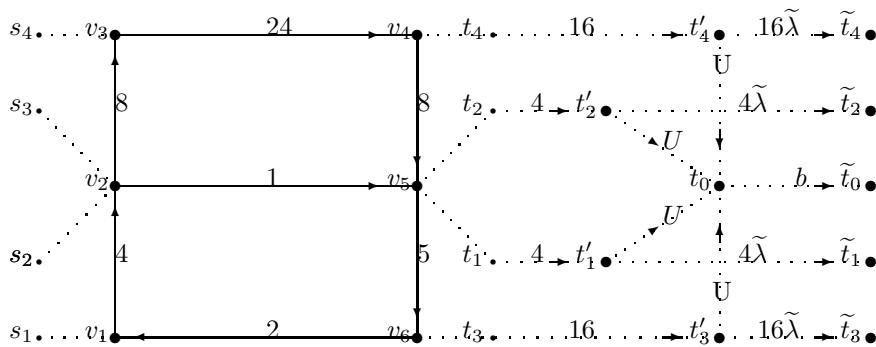


图 4 例 2 中问题 (i) 的示意图

例 2 (i) 在图 4 中, 点 v_1, v_2, v_3, v_4, v_5 和 v_6 与所有的实边一起构成一个图 G , 而所有的实边又与它们的权一起给出权函数 c . $(s_1, \widetilde{v_1}, v_1)$ 表示 v_1 是物资 1 的源, $(v_5, \widetilde{v_5t_1}, t_1)$ 表示 v_5 是物资 1 的汇, 等等. 相应地, $H = \{[s_1, t_1], [s_2, t_2], [s_3, t_3], [s_4, t_4]\}$. 综合地, G, c 和 H 形成多物资网络 $[G, c, H]$. 关于 $t_0, \tilde{t}_0, t'_i, \tilde{t}_i$ 和 b 的意义, 请回顾算法 3. 进一步, (t_i, \tilde{t}_i, t'_i) 与它的权联合表示物资 i 的需求量是 b_i . 相应地, $\mathbf{b} = (b_1, b_2, b_3, b_4) = (4, 4, 16, 16)$.

如果将每一个点 t_i 都收缩到它的对应点 v_i , 则所有的有向边与它们的端点和权一起就给出了算法 3 中辅助的 \tilde{G} 与 \tilde{c} . 相应地, $\tilde{H} = \{[s_i, \tilde{t}_i], [s_i, \tilde{t}_0] : i = 1, 2, 3, 4\}$.

关于由上述的 $[G, c, H]$ 与 \mathbf{b} 所给出的 MMFP-GFMR, 表 1 汇总给出了运用算法 3 及其它方法求解的数字结果等. 用它们可以进一步地检验我们的工作.

(ii) 类似于 (i), 针对图 5, 设 $s = [1, 1, 2, 4]$, $t = [4, 5, 3, 5]$ 和 $b = (6, 2, 6, 2)$, 则我们又得到一个 MMFP-GFMR. 关于求解该 MMFP-GFMR 的数字结果, 请见表 2.

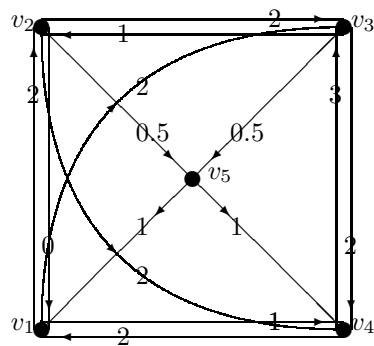


图 5 例 2 中问题 (ii) 的示意图

表 1 关于例 2 中问题 (i) 的计算结果

物资, 源与汇		$[s_1, t_1]$	$[s_2, t_2]$	$[s_3, t_3]$	$[s_4, t_4]$
需求量		4	4	16	16
一致点的分量		2.5	2.5	10	10
理论结果 1	分流量	2	2	5	16
	OPT, TSD	25		6	
理论结果 2	分流量	2.5	2.5	4	16
	OPT, TSD	25		6	
算法 3 计算 的 结果	误差参数	$\eta = 0.5$		$\varepsilon_0 = 0.02$	
	分流量	2.4014	2.4015	4.1476	15.7032
	总流量, TSD	24.6528		5.7023	
算法 2 计算 的 结果	误差参数	$\varepsilon = 0.02$			
	分流量	3.7377	3.7400	1.4491	15.9482
	总流量, TSD	24.8750		8.4259	
算法 1 计算 的 结果	误差参数	$\varepsilon = 0.02$			
	分流量	3.6545	4.6707	0	16.6216
	总流量, TSD	24.9468		9.9468	

注释 $TSD = \sum \max\{V_i(y) - \lambda b_i, 0\}$ (总上偏差), $\lambda = \frac{\bar{V}}{\sum b_i}$.

表 2 关于例 2 中问题 (ii) 的计算结果

物资, 源与汇		$[s_1, t_1]$	$[s_2, t_2]$	$[s_3, t_3]$	$[s_4, t_4]$
需求量		4	2	8	2
一致点的分量		2	1	4	1
理论结果 \bar{V} , TSD	分流量	3	0.5	4	0.5
	\bar{V} , TSD	8		1	
算法 3 计算 的 结果	误差参数	$\eta = 0.2$		$\varepsilon = 0.02$	
	分流量	2.9447	0.5	3.9986	0.5
	总流量, TSD	7.9433		0.9447	
算法 2 计算 的 结果	误差参数	$\varepsilon = 0.02$			
	分流量	3.7107	0.3736	3.4131	0.5
	总流量, TSD	7.9974		1.7107	
算法 1 计算 的 结果	误差参数	$\varepsilon = 0.02$			
	分流量	4.7784	0.0316	2.6774	0.5092
	总流量, TSD	7.9966		2.7784	

表 1 和表 2 的结果表明: 对于求解问题 MMFP-GFMR 来说, 算法 3 是很有效的, 其远远地优越于算法 1 和算法 2; 定理 4 的结论是正确的. 此外, 通过这两个实例, 我们还可以进一步地理解问题 MMFP-GFMR.

7 结束语

本文给出和研究了具有全局性公平满意度的最大多物资网络流问题. 这一模型可以做为研究与探索某些实际问题的最优决策的有利工具. 关于该问题, 我们证明了其解存在; 设计了一个求解的拟多项式逼近算法; 给出与证明了我们所设计出的算法的复杂性与逼近程度. 此外, 我们的工作还为研究网络流问题提供了一些新的思路. 最后, 我们衷心地希望本文能够推动网络流研究的发展.

致谢! 我们衷心地感谢越民义先生的指导和帮助, 衷心地感谢审稿人对本文的认真评审和指导.

参 考 文 献

- [1] Sheu J B. An Emergency Logistics Distribution Approach for Quick Response to Urgent Relief Demand in Disasters. *Transp. Res. Part E: Logist Transp. Rev.*, 2007, 43(6): 687–709
- [2] Yi W, Kumar A. Ant Colony Optimization for Disaster Relief Operations. *Transp. Res. Part E: Logist Transp. Rev.*, 2007, 43(6): 660–672
- [3] Nace D, Doan L N, Klopfenstein O, Bashllari A. Max-min Fairness in Multi-commodity Flows. *Comput. Oper. Res.*, 2008, 35(2): 557–573
- [4] Korte B, Vygen J. Combinatorial Optimization Theory and Algorithms. Berlin: Springer-Verlag, 2000
- [5] Cheng C D, Li Z P. Improvements on the Proof of an Approximate Scheme for the Maximum Multi-commodity Flow Problem. Proceedings of the 4th International Conference on Intelligent Information Technology Application, Vol.II, 2010, 311–314
- [6] Shahrokhi F, Matula D W. The Maximum Concurrent Flow Problem. *J. Assoc. Comput. Math.*, 1990, 37(2): 318–334
- [7] Garg N, Könemann J. Faster and Simpler Algorithms for Multicommodity Flow and Other Fractional Packing Problems. Proceedings of the 39th Annual IEEE Symposium on Foundations of Computer Science, 1998, 300–309
- [8] Fleischer L K. Approximation Fractional Multicommodity Flow Independent of the Number of Commodities. *SIAM J. Discrete Math.*, 2000, 13(4): 505–520
- [9] 刘振宏, 蔡茂诚 (译). 组合最优化算法和复杂性. 北京: 清华大学出版社, 1988
(Liu Z H, Cai M C (translate). Combinatorial Optimization Algorithms and Complexity. Beijing: Tsinghua University Press, 1988)
- [10] 越民义 (译). 芯片设计—组合最优化的特殊应用. 上海: 上海科学技术出版社, 2009

- (Yue M Y (translate). How Does One Design a Chip—a Specific Application of Combinatorial Optimization. Shanghai: Shanghai Scientific & Technological Press, 2009)
- [11] 谢金星, 邢文训, 王振波. 网络优化 (第二版). 北京: 清华大学出版社, 2009
(Xie J X, Xing W X, Wang Z B. Optimization on Network (second edition). Beijing: Tsinghua University Press, 2009)

Maximum Multicommodity Flow Problem with Global Fair Met Rate

CHENG CONGDIAN

(College of Mathematics and Systems Science, Shenyang Normal University, Shenyang 110034)

(E-mail: zhiyang918@163.com)

LI ZHENPENG

(Institute of Systems Science, Academy of Mathematics and Systems Science,

Chinese Academy of Sciences, Beijing 100190)

(E-mail: lnlzhp2006@qq.com)

Abstract The present work studies a maximum multicommodity flow problem with global fair met rate, which not only enriches the content of the maximum multicommodity flow problem, but also can be used to study the optimization decisions of some practical problems, such as the operations of logistics distribution stemming out of transportation. The major contributions are as follows: (A) Propose the problem and prove its solutions exists. (B) Design a pseudopolynomial approximation algorithm to solve the problem. (C) Propose and prove an approximation measure of the designed algorithm. (D) Finally, computational experiments with two instances are conducted using the algorithm.

Key words network; multicommodity flow; logistics distribution; optimization decision; approximation algorithm; approximation measure

MR(2000) Subject Classification 90C35; 90C27

Chinese Library Classification O221.7