多机器人任务分配问题的分布式算法

Stefano Giordani[1] , Marin Lujak[1] , and Francesco Martinelli[2]

[1]硕士。意大利罗马大学 "Tor Vergata "的Ingegneria dell'Impresa课程。
[2]硕士生导师。意大利罗马大学 "Tor Vergata "分校信息系统与生产部，意大利

摘要。在这项工作中，我们解决了多机器人任务分配问题（MRTA）。我们假设决策环境是分散的，决策人（代理）的数量与系统中的机器人一样多。为了解决这个问题，我们为分配问题开发了一个分布式的匈牙利方法。机器人在个人的基础上，以及通过系统中其他机器人的传话收到的信息，自动执行匈牙利算法的不同子步骤。假设每个机器人代理都有关于它与环境中目标的距离的信息。机器人之间的通信是通过一个连接的动态通信网络进行的，分配问题的解决方案是在没有任何共同的协调者或系统的共享存储器的情况下达成的。该算法在$O(n^3)$的累积时间内(每个机器人为$O(n^2)$)得出了一个全局最优的解决方案，N个机器人之间交换的信息数量为$O(n^3)$。

关键词。多机器人任务分配，分配问题，分布式算法。

1 简介

在这项工作中，我们考虑了多机器人任务分配问题（MRTA），其目标是根据一些全局目标函数的优化，找到n个机器人对n个任务集的分配（例如，见Gerkey和Mataric，2003）。MRTA对应于（线性和）分配问题，最早开发的算法是匈牙利法（见Kuhn, 1955）。

匈牙利方法或Kuhn-Munkres算法是一种众所周知的迭代算法，它在计算过程中保持对偶可行性，并寻找满足互补松弛条件的原始解。如果原始解是可行的，该解就是最优的。如果原始解不可行，该方法会对二元可行解进行修改，然后开始新的迭代。匈牙利方法可以使用交替树来实现，因此其最坏情况下的时间复杂度被限制为$O(n^3)$（例如，见Papadimitriou和Steiglitz 1982）。

我们假设决策环境是分散的，决策人（代理）的数量与系统中的机器人一样多。为了解决这个问题，我们开发了一个分布式的匈牙利方法。机器人

N.Garc´ıa-Pedrajas et al. (Eds.):IEA/AIE 2010, Part I, LNAI 6096, pp.721-730, 2010.

在个人和通过系统中其他机器人的信息收到的基础上，自主地执行匈牙利算法的不同子步骤。假设每个机器人代理都有关于它与环境中的目标的距离的信息，或者，更广泛地说，有相同的分配成本。机器人之间的通信是通过一个连接的动态通信网络进行的。该算法在$O(n^3)$的累积时间内（每个机器人为$O(n^2)$）得出一个全局最优解，在N*个机器人*之间交换的信息数为$O(n^3)$。各个处理器之间没有共享的内存，在没有任何系统的共同协调者的情况下，就能达成对分配问题的解决。任务是无记忆的，实际上被视为没有任何计算或记忆能力的对象。

本文的大纲如下。在第2节，我们介绍了相关工作。在第3节中，我们阐述了一些基本的定义以及问题的表述。在第4节中，介绍了分布式分配算法。在第5节中，我们介绍了仿真结果。最后，我们以未来工作的可能方向和第6节的结论来结束本文。

2 相关工作

有几种主要的方法来解决赋值问题（例如，见Burkard和Çela，1999）。

经典的集中分配方法通过对某些成本函数的迭代改进来找到匹配的解决方案：在原始单线方法中，它是一个原始成本，而在匈牙利、双单线和放松方法中，它是一个双成本（见Bertsekas，1992）。

拍卖算法可以通过中间迭代改善或恶化原始成本和对等成本，尽管最后找到了最优分配（见Bertsekas，1991）。Bertsekas在这项工作中介绍了拍卖算法，其中代理人以迭代的方式竞标任务，在每次迭代中，竞标增量总是至少等于E（E-）。
补充松弛度）。如果$E < \frac{1}{n}$，算法就能找到最优解。
并在$O(n^3 \cdot \max\{c_{ij}\})$时间内运行。Zavlanos等人（2008）提供了一个分布式的Bertsekas为网络系统提出的拍卖算法的版本
由于代理人的通信能力有限，缺乏全球信息。准确竞价所需的最新价格只能通过相邻代理之间的本地信息交流以多跳方式获得。没有共享内存，代理人需要在本地存储所有的价格信息。这种方法计算出的最优方案在
$O(\Delta \cdot n^3 \cdot \max\{c_{ij}\})$时间，其中$\Delta \leq n-1$为最大网络直径
的通信网络。

也有许多基于匈牙利方法的并行算法。一个很好的调查见，例如，Burkard和Çela, 1999, Bertsekas等人, 1995。

另一种看待分配问题的方式是通过博弈论的表述（见，例如，Arslan等人，2007）。

在机器人界，有许多针对MRTA问题的启发式方法。Smith和Bullo（2007）描述了两种算法，即ETSP和网格算法，用于稀疏和密集环境中的任务分配。代理人对环境中的任务有充分的了解，在ETSP算法中，他们接近最近的任务。如果最近的任务被占用，他们会预先计算出一个通过其余n个任务的最佳行程，并根据某些标准搜索第一个未被占用的任务。Gerkey和Mataric（2003，2004）通过三个重要因素描述了MRTA问题的某些相关启发式算法：计算和通信要求，以及考虑（重新）分配任务的方式，即在每次迭代中考虑（重新）分配全部还是部分任务。这些方法的问题是，正如研究者所指出的那样，没有对算法可预期的解决方案的质量进行描述。Kwok等人（2002）将经典的组合方法，如匈牙利算法和Gabow算法，应用于一组移动机器人的分配问题，这些机器人需要移动到所需的网格点矩阵中，以便对所需区域进行全面监视。

自上而下的集中式方法，以及一般的集中式控制，在受控部件数量增加的情况下是非常复杂和昂贵的，在我们的例子中是机器人，并且受到资源限制、性能瓶颈和关键故障的影响（例如，见Sugar和Kumar，2000）。相反，自下而上的多代理模块化方法中的协调系统将计算资源和能力分配到相互连接的代理网络中，因此降低了复杂性并提高了相同的稳健性。多代理系统不存在与集中式系统相关的 "单点故障 "问题（见Wooldridge, 2002）。此外，在未知的动态环境或未知的移动障碍物的情况下，就像有移动生产机器人的工厂和随机需求的情况一样（见，例如，Giordani, Lujak, and Martinelli, 2009），局部规划比全局规划更受欢迎。如果系统中存在许多代理，或者存在不同的需求和能够满足需求的多种生产机器组合，那么后者就很昂贵。与集中式算法相比，该算法的分散式实施的一些其他优点是。(例如，见Lueth和Laengle 1994）模块化：机器人在本质上是独立的；分散的知识库：每个机器人使用一个本地知识库，存储机器人的相关信息，并能够与其他机器人交换这些信息；容错和冗余：在单个机器人出错的情况下，多机器人系统继续运行，以及可扩展性：新的机器人可以被添加到原始系统中，而无需改变系统结构。

3 定义和问题的提出

我们认为一个系统由一组R*组成，其中有*n*个*协作的移动机器人代理，在一个环境中，有n个目标地点（任务）的集合T和一个
给定$n \times n$矩阵的机器人之间的距离（或成本）c_{ij}。

每个机器人可以被分配到最多一个任务，每个任务可以由不超过一个机器人执行。该问题包括找出机器人和任务之间总成本（距离）最小的可行分配，即确定

的最小权重完美匹配的加权双子图$G = (T \cup R, E)$，其权重c_{ij} 在边（i, j）$\in E$上问题是。

$$\min \sum_{i,j} c_{ij} x_{ij} \tag{1}$$

受制于

$$\sum_{i=1}^{n} x_{ij} = 1, \ \forall j \ , \tag{2}$$

和

$$\sum_{j=1}^{n} x_{ij} = 1, \ \forall i \text{。} \tag{3}$$

$$x_{ij} \geq 0 \text{。} \tag{4}$$

虽然存在分数解，但总是存在一个最佳的整数解，它对应于一个真实的匹配（例如，见Papadimitriou和Steiglitz，1982）。特别是，$x_{ij} = 1$，如果机器人j与任务i匹配；否则$x_{ij} = 0$。

将上述问题的表述称为原始问题，就可以将对偶问题定义如下。让α_i 是与任务i相关的对偶变量，β_j 是与机器人j相关的对偶变量。这两个变量都是实数，且符号不受限制。对偶问题的目标是

$$\max_{j} \{\sum_{i} \beta_j - \sum \alpha\}_i \tag{5}$$

受制于

$$\beta_j - \alpha_i \leq c_{ij}, \forall i, \forall j \ . \tag{6}$$

可以对对偶问题和对偶变量进行经济解释：α_i 是每个机器人j在与任务i匹配时将支付的价格，而β_j 是机器人j与某个任务匹配时的效用。对偶问题的约束条件（6）规定，机器人j的效用β_j 不能大于分配（匹配）到任务i的总成本（$c_{ij} + \alpha_i$）。要最大化的对偶目标函数（5）是机器人的效用之和与任务的价格之和之间的差值，也就是

机器人的总净利润。在线性规划的二元性基础上。

$\sum_j \beta_j - \sum_i \alpha_i \leq \sum_{ij} c_{ij} - x_{ij}$ ；因此，机器人的总净利润不能大于机器人为移动所需支付的总分配成本。

到任务，而且只有在最佳状态下，这两者才是相等的。显然，在最佳状态下，每个机器人j将被匹配到任务i上，机器人的效用为$\beta_j = c_{ij} + \alpha_i$，也就是说，正好等于机器人$j$与任务$i$匹配的总成本。

4 分布式分配算法

假设每个机器人代理都有关于其位置的信息，并且可以通过环境地图上的坐标接收关于环境中所有任务的位置信息。机器人之间的通信是通过一个连接的动态通信网络进行的，分配问题的解决不需要任何共同的协调者或系统的共享存储器。

为了解决这个问题，我们开发了一个分布式的匈牙利方法，该方法具有图论中的增强路径，我们将在下文中简要地回顾。

我们指的是通过所谓的交替树来实现的集中式（匈牙利）方法（见，例如，Burkard和Çela，1999）。该算法

该算法是迭代式的，在保持可行的对偶解的情况下，考虑可接受的双边图$\bar{G}=(T \cup R, \bar{E})$，其中$\bar{E}=\{(i,j) : c_{ij}+\alpha_i-\beta_j=0\}$。该算法在图$\bar{G}$中寻找一个最大cardinality的匹配。如果
匹配是完美的，也就是说，每个机器人都与一个任务相匹配，那么匹配-------。
ing代表最优解，算法停止。如果匹配不完美，算法就会更新对偶变量，以增加对偶目标函数，使$\bar{G}$中至少增加一条新的可接受边，并且
继续进行新的迭代。

给定$\bar{G}$，让$M\subseteq\bar{E}$是$\bar{G}$中当前的最大匹配。边缘
属于M的边被称为匹配边，$\bar{E}\setminus M$中的边是自由边。给出$\bar{G}$和当前的最大匹配$M\subseteq\bar{E}$，该算法沿着$\bar{G}$中的交替树上的增强路径迭代地改进匹配。交替树是$\bar{G}$的一个子图，它以某个自由任务顶点t为根，通过$\bar{E}$的自由边与所有相邻的机器人顶点相连，并且每个机器人
通过一条匹配的边与一个任务顶点相连。当上述
提到的树到达一个任务叶顶点，该顶点与某个自由机器人顶点r相邻，一个增强路径被找到，即一个允许交换自由边和匹配边的增强匹配度的路径。当所有可能的增强步骤都完成后，对偶变量被更新，新的可接受边被添加。然后，一个新的迭代开始在新的可接受图上搜索最大匹配（例如，见Burkard和Çela，1999）。

在我们的实现中，我们维护所有根植于自由任务顶点的交替树的森林F^1。此外，我们还维护了一个交替树的森林F。[2]
$\bar{G}$中以机器人顶点为根的树，包含所有不属于机器人任务的顶点。
包含在F^1。显然，后者也不与F^1中的顶点相连。

最初，森林是按以下方式设置的。森林F^1是由n个孤立的任务顶点组成的。森林F^2，以类似的方式，由n个孤立的机器人顶点组成。对于所有的任务i，双变量被假定为$\alpha_i=0$，并且
$\beta_j=\min_i c_{ij}$，对于所有的机器人j，并按以下方式更新。设$minSlack_j=\min_{i\in F^1}(c_{ij}+\alpha_i-\beta_j)$，$n_j$是对应arg min的任务，为

每个机器人顶点$j \in F^2$，而$\delta = \min_{j \in F^2} minSlack_j$。对偶变量的新值为：。

$$\alpha_i := \alpha_i - \delta, \forall \text{任务顶点} i \in F^1$$
$$\beta_j := \beta_j - \delta, \quad \forall \text{机器人顶点} j \in F^1$$

在更新对偶变量之后，即使可能有不止一条新的可接受边，w.l.o.g.，我们也可以一次添加新的边，在只添加一条边之后，在增强的可接受图上搜索新的匹配。新的边连接着两个森林。更详细地说，这条边开始于
在F^1的任务顶点t^*，在F^2的机器人顶点r^*结束。如果r^*是自由的（未匹配），那么就有一条增强路径（$t, \ldots, t^*, r^*$），从F^1的根任务顶点开始，通过交替路径与t^*相连，并在r^*结束。通过交换这个增强路径中的自由和匹配边
路径，我们得到增强的匹配。由于任务顶点t不再是自由的，F^1中以t为根的所有树都移动到F^2，并通过t^*与根r^*相连。如果r^*不是自由的，即它被匹配，就没有增强路径，两个交替树的森林被更新，将机器人r^*及其所有子树从森林F^2中断开，并将其所有子树与森林中的任务顶点t^*相连。
F^1。

4.1 算法细节

机器人根据本地信息和通过通信图中的邻居的信息收到的信息，交互地执行算法的步骤。通信图的结构是由限制于机器人的森林F^1和F^2组成。每次森林发生变化时，相互连接的机器人会重新计算森林的结构。因此，每次森林F^1和F^2的更新发生时，连接图都会更新。

如图1所示，每个机器人在其内存中保留了4个与树上邻近位置的机器人有关的指针，即父亲机器人、长子、下一个哥哥和下一个弟弟。前两个点是用来在树中向上和向下移动的，而后两个点是用来探索一个分支的。每个机器人在其内存中还保留了关于其位置（是否在F^1或F^2）、其匹配的任务M_j（如果没有，$M_j = 0$）和父亲任务顶点的数据。每个机器人j也会记住距离矩阵的列向量j，即它的位置到所有任务位置的距离向量，它的对偶变量β_j，以及$minSlack_j$，以及获得$minSlack$的任务n_{jj}。

通过自主计算和与邻居的交流，机器人获得并分享所有任务i的α_i、δ值的对偶变量更新、F^1中的任务、森林F^1和F^2之间的新边，以及森林F^1和F^2的根机器人$r(F^1)$和$r(F^2)$（森林深度搜索中的第一个机器人顶点）的信息。这样一来，机器人系统就没有共同的协调者或共享存储器。具体来说，机器人根据在森林中的位置（例如，他们是否在增殖路径中，在森林F^1或F^2，等等）改变他们的角色。

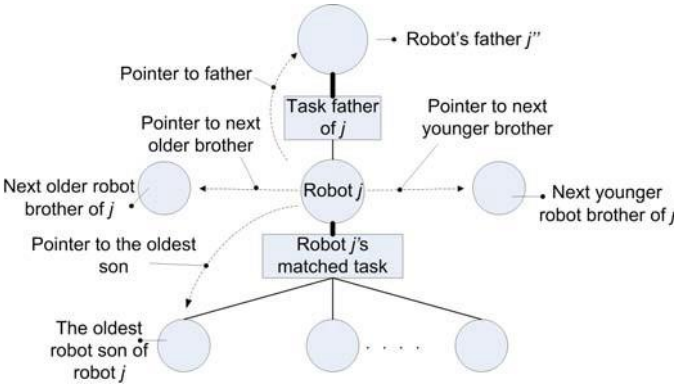

图1.每个机器人在树上的四个指针

并相应地完成分布式算法的一些步骤。在下文中，我们根据机器人的角色来描述它们的行动。

初始化（由每个机器人j完成）。

- 让M_j:=0（机器人j最初未被匹配）；让α_i:=0为所有任务i，β_j:=$\min_i\{c_{ij}\}$。
- 以这样的方式初始化指针，即机器人j只与它的弟弟$(j-1)$和哥哥机器人$(j+1)$相连，没有父亲，也没有儿子。这意味着森林被初始化，如上所述。
- 每个定位在F^2的机器人计算$[minSlack_j, n_j]$，而不与其他机器人交换任何信息。

根机器人$r(F^1)$和$r(F^2)$开始基于深度搜索（DFS）的信息交换，询问是否有未匹配的机器人。在与第一个不匹配的机器人接触时，同样通知$r(F^2)$开始一个新的迭代。在下文中，我们将描述机器人在一次迭代中的角色行动。

- 所有的机器人$j\in F^2$都参与计算$[\delta,(t^*,r^*)]$。δ的计算从根机器人$r(F^2)$开始，并通过F^2的第一深度搜索进行信息交换。当找到$[\delta,(t^*,r^*)]$的值时，信息将通过消息传递给F^2的所有机器人。

  基于深度搜索（DFS）的交换，从机器人的根部开始。F^2。同样的DFS消息传递也适用于通知F^1中的机器人。
- 所有的机器人j做以下事情：对于所有$t_i\in F^1$设置α_i:=$\alpha_i-\delta$；此外，如果j属于F^1，设β_j:=$\beta_j-\delta$。在这个阶段没有信息交换。
  - t^*（如果有的话）的父亲机器人，返回关于他自己的信息和关于他的大儿子，通过DFS消息传递给所有机器人。

- 根机器人$r(F^2)$通知F^2中的r^*，开始下一阶段的扩增或不扩增路径。
- 如果r^*没有被匹配，那么它就开始增殖。通过r^*和增殖路径中的机器人之间的信息传递，机器人r^*命令交换增殖路径上的自由边和匹配边。它发送了
  信息给F^2中的机器人，并提供与F^1中任务t^*子树的新连接信息。该信息包含t^*，以及F^1中所有向后的对象，直到沿着增强路径（$t, \ldots, t^*, r^*$）的第一个无父任务。机器人同时更新关于从r^*收到的信息的匹配，并更新相邻机器人的指针，以便F^1中以t移动为根的t^*的树。
  到F^2，并通过t^*连接到根r^*。机器人通过DFS
  信息，以获得有关该组织位置的最新信息。
  任务在森林方面。定位在F^2的每个机器人计算新的$[minSlack_j, n_j]$，而不与其他机器人交换任何信息。
- 如果r^*已经被匹配，那么它就不会增强匹配。它将发送
  向$r(F^2)$发送消息，通过DFS消息传递，通知F^2中的所有机器人将最小松弛量减少δ；r^*从F^2树上断开连接，将自己和它的子树通过t^*连接到F^1。F^2中的每个机器人为r^*路由的子树上的每个任务更新新的最小松弛量。
  F^1，这导致了$O(n^2)$消息的交换。

关于该方法的稳健性，如果机器人在算法执行过程中停止响应，它就会被认为是错误的，并在进一步的计算中被淘汰。如果机器人在森林F^2中没有被匹配，则计算继续进行，不做任何修改，忽略有关的机器人。否则，算法将从头开始，不考虑同样的情况。

5 仿真结果

仿真是用C语言进行的，在一台装有Intel Core Duo 1.6 GHz CPU和1GB内存的电脑上执行。对于从10到250的每个可能的n，我们考虑了10个不同的实例的平均值，包括
稀疏的随机成本矩阵，稀疏度从2到10%不等，成本$c_{ij} \in (0, 100)$。
从实验中可以看出，在一个人的计算时间中，有一个人的累计计算时间超过了10分钟。
由于机器人之间的信息交换，分散式方法的计算时间比集中式算法的计算时间高一到两个数量级，但仍然受到$O(n^3)$时间的限制（见图2，左）。机器人之间交换的信息数量也受到$O(n^3)$的限制（见图2，中间）。即使分散的方法累积起来更重，每个机器人执行的计算时间（在$O(n^2)$的数量级）也比标准（集中式）匈牙利算法所需的时间小（见图2，右）。

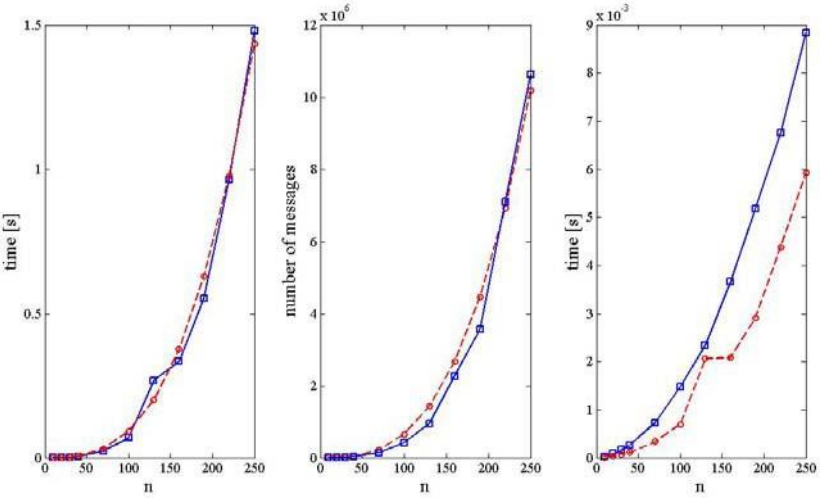

图2.左边：分散算法的累积时间（实线）与n^3 最佳fi t - ting曲线（虚线）；中间：信息总数（实线）与n^3 最佳fiting曲线（虚线）；右边：（集中的）匈牙利算法的计算时间（实线）与分散方法中每个机器人的计算时间（虚线）。

6 总结

我们提出了匈牙利算法的分散实现，以解决一组n个机器人的分配问题，这些机器人必须在没有集中式控制器和共享存储器的分散结构框架内执行一定数量的任务。这些机器人必须自主运行，并且在信息交换的基础上可以实现分散化的实施。我们假设机器人和 任务的数量相等，但更普遍的情况可以通过引入具有精确成本的精确机器人/任务来轻松处理；同时，机器人只能执行任务的一个子集的情况也可以通过适当选择矩阵成本来处理。分散式算法的累积执行时间和交换的信息总数都在$O(n^3)$的数量级，从而使每个机器人的平均计算载荷为$O(n^2)$。分散式的实施在本质上是稳健的：介绍了如何处理机器人故障，并将在未来的研究中进行扩展。

参考文献

Arslan, G., Marden, J.R., Shamma, J.S.:自主车辆-目标分配。一个博弈理论的表述。Journal of Dynamic Systems, Measurement, and Con-trol 129, 584-596 (2007)

Bertsekas, D.P.:Linear Network Optimization:算法和代码。麻省理工学院出版社，剑桥(1991)

Bertsekas, D.P.:网络流问题的拍卖算法。A tutorial introduction.
计算优化和应用1(1) (1992)

Bertsekas, D.P., Castanon, D.A., Eckstein, J., Zenios, S.: 网络工作优化的并行计算。In: 网络模型-运筹学和管理科学手册》，第7卷，第330-399页。Elsevier, Amsterdam (1995)

Burkard, R.E., Cęla, E.: Linear assignment problems and extensions.组合优化手册》4(1), 221-300 (1999)

Gerkey, B.P., Mataric, M.J.:一个研究多机器人任务分配的框架。在。Shultz, A.C., et al. (eds.) Multi-Robot Systems:From Swarms to Intelligent Au- tomata, The Netherlands, vol. 2, pp.Kluwer Academic Publishers, Dordrecht (2003)

Gerkey, B.P., Mataric, M.J. 。多机器人系统中任务分配的正式分析和分类法。
International Journal of Robotics Research 23(9), 939-954 (2004) Giordani, S., Lujak, M., Martinelli, F.: A Decentralized Scheduling Policy for a Dynam-School.
可重构的生产系统。在。Maˇrˇk, V., Strasser, T., Zoitl, A. (eds.) Holonic and Multi-Agent Systems for Manufacturing.LNCS（LNAI），第5696卷，第102-113页。斯普林格，海德堡 (2009)

库恩，H.W.：分配问题的匈牙利方法。海军研究后勤季刊》2, 83-97 (1955)

Kwok, K.S., Driessen, B.J., Phillips, C.A., Tovey, C.A.: 使用最优分配算法分析多目标-多代理情景。智能和机器人系统杂志》35(1), 111-122 (2002)

Lawler, E.L.: Combinatorial Optimization: 网络和Matroids。Holt, Rinehart, and Winston (1976)

Lueth, T.C., Laengle, T.: 分布式自主多Agent机器人系统的任务描述、分解和分配。In:IEEE/RSJ IROS，第1516-1523页(1994)

Papadimitriou, C.H., Steiglitz, K.: Combinatorial optimization: algorithms and complexity.Prentice-Hall, Inc., Englewood Cliffs (1982)

Smith, S.L., Bullo, F.: 机器人网络的目标分配。有限通信下的渐进性能。In:美国控制会议，ACC'07，第1155-1160页（2007）。

Sugar, T., Kumar, V.: 多个移动机器人在操纵和材料处理任务中的控制和协调。In:第六届实验机器人学国际研讨会，第六卷，第15-24页。Springer, Heidelberg (2000)

Wooldridge, M.: Introduction to Multi-Agent Systems.John Wiley and Sons, Chichester (2002)

Zavlanos, M.M., Spesivtsev, L., Pappas, G.J.:分配问题的分布式拍卖算法。In:Proc. of 47th IEEE Conf. on Decision and Control, Cancun, Mexico, pp.1212-1217 (2008)