

何杏宇, 吴西博, 杨桂松, 等. 面向多类型资源需求的机器人任务分配方法[J]. 智能计算机与应用, 2024, 14(6): 11-19.
DOI:10.20169/j.issn.2095-2163.240602

面向多类型资源需求的机器人任务分配方法

何杏宇^{1,2}, 吴西博¹, 杨桂松¹, 贾明权³

(1 上海理工大学 光电信息与计算机工程学院, 上海 200093; 2 上海理工大学 出版印刷与艺术设计学院, 上海 200093;

3 中国西南电子技术研究所, 成都 610036)

摘要: 针对多机器人系统中任务的多类型资源需求问题, 单个机器人往往因能力有限而无法胜任, 需要多个机器人来协同完成。为此, 提出了一种面向多类型资源需求的任务分配优化模型, 在单个机器人携带资源有限的情况下, 为任务分配合适的机器人协同小组。该模型综合考虑了影响任务协同成本的时间、移动距离、资源利用率等因素, 设计了任务的协同成本函数, 并提出基于纳什均衡策略的任务分配方法。该方法首先通过遗传算法为更多的任务分配符合其资源需求的初始机器人小组, 然后利用博弈论中的纳什均衡思想为初始机器人小组寻找最优平衡解, 最终得到满足纳什均衡解的机器人小组, 从而降低任务的协同成本。实验结果表明, 本文所提方法能有效提高任务完成率和降低任务的平均协同成本。

关键词: 多机器人系统; 任务分配; 协同小组; 遗传算法; 纳什均衡解

中图分类号: TP393

文献标志码: A

文章编号: 2095-2163(2024)06-0011-09

A robot task allocation method for multi-type resource requirements

HE Xingyu^{1,2}, WU Xibo¹, YANG Guisong¹, JIA Mingquan³

(1 School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, Shanghai 200093, China; 2 College of Communication and Art Design, University of Shanghai for Science and Technology, Shanghai 200093, China; 3 Southwest China Institute of Electronic Technology, Chengdu 610036, China)

Abstract: In view of the multi-type resource demand of tasks in multi-robot systems, individual robots are often unable to perform the task by their limited abilities, so that multiple robots are required to complete cooperatively. To address the problem, this paper proposes a task allocation optimization model for multiple resource requirements to assign the task to a suitable robot cooperative group with individual robot limited resources. This model designs the collaboration cost function of the task which comprehensively considers time, travel distance, resource utilization and other factors affecting task collaboration cost. And the model proposes a task assignment method based on the Nash equilibrium strategy. Firstly, this method assigns more tasks to initial robot groups that met their resource needs by Genetic Algorithm. Then, the optimal equilibrium solution is achieved for the initial groups by Nash equilibrium thought in Game Theory. Finally, this method obtains the robot groups that meets the Nash equilibrium solution, so as to reduce the collaborative cost of tasks. The experimental results show that the proposed method can effectively improve the task completion rate and reduce the average collaborative cost of the task.

Key words: multiple robot systems; task allocation; collaborative group; Genetic Algorithm; Nash equilibrium solution

0 引言

近些年来, 一些智能化、便捷化的设备不断改变着传统的生活方式。特别是, 随着无线通信、大数据和人工智能技术的发展, 机器人系统逐渐走向成熟。

其中, 机器人凭借着机动性和灵活性等优势, 可以完成许多复杂、重复性和危险环境下的工作, 现已广泛应用在许多领域, 如现代军事^[1]、物资运输^[2-3]、智慧城市^[4-5]、应急救援^[6-7]等方面。然而, 由于不同领域中的任务环境日益复杂, 单个机器人在续航、信

基金项目: 国家自然科学基金(61802257, 61602305); 上海市自然科学基金(18ZR1426000, 19ZR1477600); 敏捷智能计算四川省重点实验室开放式基金资助项目。

作者简介: 何杏宇(1984-), 女, 博士, 副教授, 硕士生导师, 主要研究方向: 物联网, 群智计算大数据分析; 吴西博(1996-), 男, 硕士研究生, 主要研究方向: 无人自主系统, 群智计算; 贾明权(1982-), 男, 博士, 高级工程师, 主要研究方向: 先进智能计算。

通讯作者: 杨桂松(1982-), 男, 博士, 副教授, 硕士生导师, 主要研究方向: 物联网, 普适计算。Email: gsyang@usst.edu.cn

收稿日期: 2023-04-08

息的获取和计算等方面都是有限的,很难单独完成任务要求,需要多个机器人协同完成复杂的任务。多机器人系统任务分配是机器人系统研究的关键问题,任务分配算法的好坏直接影响着系统任务的执行效率和成本。

多机器人系统中针对资源需求的分配场景是任务分配研究的一个重要方面,如应急救援场景中各种物资资源分配、智慧医疗中医疗机器人资源的智能化调度、城市环境监管中的传感器资源分配等。在这些场景中任务大多具有异构和多种类型资源需求的属性,而单个机器人的执行能力有限,需要多个机器人来协作完成,导致机器人与任务需求之间具有复杂的匹配关系。因此,在系统机器人资源有限的情况下,如何分配有限的资源来高效完成任务显得至关重要。

现有针对资源需求的任务分配研究中,主要分为单类型资源需求和多类型资源需求的2种任务场景。在任务单类型资源需求^[8-9]的场景中,任务的资源需求都是单一类型且同构的,每个机器人都具备执行任务的能力,系统根据不同的场景要求,通过设计相应的任务分配算法使机器人高效地完成任务。在任务是多类型资源需求的场景中,机器人在执行任务时需要考虑多方面的因素,如任务完成时间、资源利用率、能量消耗等。文献[10]研究了多类型资源的分配问题,提出了一种基于博弈论和强化学习的协同资源分配解决方法,使资源公平地分配给用户,以提高资源的利用率。文献[11]通过分布式方法进行灾后救援场景中的物资资源分配,首先每个机器人使用蚁群方法构建一个任务执行序列,然后通过共识机制来解决任务序列中的任务冲突问题,进而优化受害者的营救时间。为了实现机器人类型、资源与任务需求的精准匹配,文献[12]提出了一种基于交叉熵的分配方法。该方法从候选解中抽取随机样本,然后用其更新分配概率矩阵。为了动态适应任务环境的变化,文献[13]提出了一种基于基尼系数的方法,以防止一些机器人的能量消耗过快,造成后续任务无法执行的情况。文献[14]提出了一个基于优化的任务优先级和执行框架,该框架为异构多机器人系统实现了一种具有弹性和能量感知的任务分配策略,从而使任务执行时的能量最小化。文献[15-16]研究了带有截止日期的搜救任务,通过建立相应的任务分配模型来为每个机器人规划一条最优的任务执行路径,同时优化每个任务的执行时间。文献[17-19]研究了任务分

配时机器人联盟的形成问题,通过启发式算法或者基于市场机制的方法来建立分配模型,目的是使联盟形成的总成本最小。

然而,上述基于资源需求的任务分配方法主要关注任务完成时间或者能量消耗这单一优化目标,未充分考虑机器人的资源利用率、能量消耗和任务完成时间对系统长期性能的综合影响,导致系统的长期资源利用率和任务完成率低。本文研究了任务多类型资源需求的分配场景,每个任务有多种类型的资源需求,同时任务执行时间受截止时间的约束,在单个机器人携带的资源和能量有限的情况下,为任务分配一个合适的机器人小组,以优化任务的协同成本。该任务分配场景的难点在于:在为任务分配机器人小组时不仅要优化任务的完成时间,还要兼顾小组内机器人的资源利用率和能量消耗,从而导致任务和机器人之间具有复杂的资源匹配关系。为此,本文提出了基于纳什均衡策略的任务协同小组分配方法,在满足任务需求的前提下保证系统任务执行时所花费的成本最低。本文的主要贡献如下:

(1)为了衡量协同小组分配的好坏,本文设计了任务的协同成本函数,根据任务的完成时间、资源利用率和机器人的能量消耗来综合评估小组的协同成本。

(2)为了降低任务的协同成本,建立了任务的最小化协同成本优化模型,并设计了基于博弈论思想的任务分配方法。该方法首先使用遗传算法选择满足任务资源需求的初始小组,然后通过对各小组中的机器人进行交换分配以进一步优化协同成本,最终得到满足纳什均衡解的分配小组。

1 系统模型

本文在系统模型设计时考虑了一种常见的应用场景,该场景包括一个机器人系统平台、多个任务发布者以及多个机器人,系统模型如图1所示。首先,任务发布者向平台发布多个将要执行的任务,平台根据任务的信息(包含任务资源需求的类型和数量、截止时间、位置坐标等)进行任务分配,为每个任务选择最优的机器人协同小组并将分配结果发送给相关的机器人,然后机器人按照分配结果移动到任务位置执行任务,最后机器人小组协作完成分配的任务并将自身的信息(资源剩余情况、位置坐标)反馈给平台以便进行下一次分配。由于系统中要执行的任务较多,且机器人资源有限,合理的分配策略能优化系统任务的执行成本,进而提高任务的完成效率。

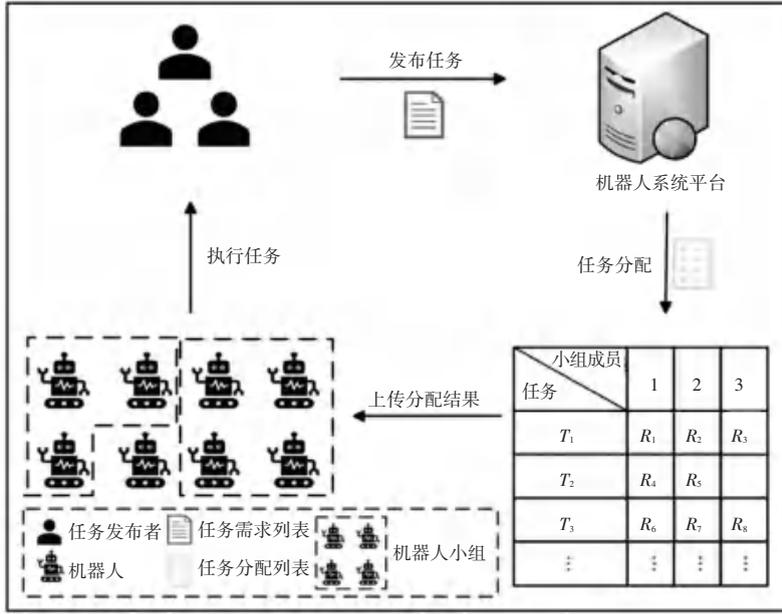


图 1 系统模型图

Fig. 1 System model diagram

1.1 问题描述

任务信息描述:假设系统发布一系列将要执行的任务, $T = \{T_1, T_2, \dots, T_k\}$ 表示系统中的任务集合, 每一个任务表示为 $T_j = \{r_j, p_j(x, y), de_j\}$ 。其中, $r_j = \{r_j(1), r_j(2), r_j(3), \dots, r_j(m)\}$ 为任务的资源需求信息, $r_j(i)$ 为任务 T_j 对 i 类型资源的需求数量, m 为系统总的资源类型数量, $p_j(x, y)$ 为任务的位置坐标, de_j 为任务执行的截止时间。任务的每种类型资源需求必须全部得到满足, 并且机器人协作组的执行时间在截止时间之前该任务才算成功完成。

机器人信息描述: $R = \{R_1, R_2, R_3, \dots, R_n\}$ 表示系统中有 n 个机器人, 每一个机器人表示为 $R_i = \{c_i, p_i(x, y), v_i, e_i\}$, $c_i = \{c_i(1), c_i(2), c_i(3), \dots, c_i(m)\}$ 为机器人 R_i 携带的资源信息。其中, $c_i(j)$ 为 R_i 携带 j 类型资源的数量, 如果 R_i 携带 j 类型资源 N 个, 则 $c_i(j) = N$, 否则 $c_i(j) = 0$ 。 $p_i(x, y)$ 表示机器人的当前位置, e_i 表示机器人当前剩余的能量, v_i 表示机器人的移动速度。由于单个机器人的执行能力有限, 每个机器人不能同时携带全部类型的资源, 只能拥有其中的一种或者几种, 机器人在初始执行任务时都会根据自身的能力携带最大容量的资源, 本文不考虑机器人资源消耗后再补充资源的情况。

1.2 机器人与任务需求的匹配关系

多机器人系统的任务分配场景通常从任务类型、机器人类型和任务分配实时性三个方面来分类描述。本文任务分配场景为单任务机器人(Single-

Task Robots, ST)、多机器人任务 (Multi-Robot Tasks, TR) 和任务实时性分配 (Instantaneous Assignment, IA), 即机器人每次最多执行一个任务 (ST), 而每个任务需要多个机器人来协作完成 (MR), 同时任务在资源需求得到满足时能够立即被执行 (IA)。本文研究的是任务数量多、机器人携带的资源 and 能量有限的任务分配场景, 由于机器人分布在任务环境中的不同位置, 当任务分配的协同小组符合其资源需求时, 任务的完成时间与协作组中机器人最晚到达任务位置的时间有关, 当组内所有机器人都移动任务位置时, 该任务能瞬时完成, 即忽略任务的执行时间。

为了展示机器人协同小组与任务需求的匹配关系, 假设任务 T_1 共有 3 种类型的资源需求 ($m = 3$), 总的需求信息为 $r_1 = \{9, 6, 7\}$, 截止时间 $de_1 = 3$ 。现有 4 个机器人 $R = \{R_1, R_2, R_3, R_4\}$, 每个机器人携带的资源为 $c_1 = \{0, 0, 8\}$, $c_2 = \{7, 0, 0\}$, $c_3 = \{0, 6, 0\}$ 和 $c_4 = \{8, 0, 8\}$ 。对于 3 个不同的小组 $G_{11} = \{R_1, R_2, R_3\}$, $G_{12} = \{R_1, R_3, R_4\}$, $G_{13} = \{R_2, R_3, R_4\}$ 分别执行任务 T_1 时, 将会产生不同的执行结果。例如, 经过计算发现协作组 G_{11} 中机器人达到 T_1 位置的最晚时间为 $t_{11} = 4$, 超过了任务的截止时间, 即 G_{11} 小组不能执行该任务; 协作组 G_{12} 和 G_{13} 的最晚到达时间在 T_1 的截止时间之前, 但 G_{12} 组内机器人剩余的总资源为 $c_{12} = \{8, 6, 16\}$, 不符合 T_1 的资源需求 $r_1 = \{9, 6, 7\}$ 。而 G_{13} 组内的剩余资源 $c_{13} =$

{15,6,8} 符合 r_1 的需求,进而能够执行 T_1 ,但 T_1 对 G_{13} 组内第一种类型资源的利用率(如 9/15)较低。因此,本文任务分配的目的在于选择任务开始执行时间早,且组内资源利用率较高的分配方案。

1.3 任务协同成本分析

系统平台每次根据当前任务和机器人的属性信息为任务组建最优的机器人协同小组,以降低每个任务的执行成本,提高系统任务的执行成功率。本文从协同小组执行任务时所花费的时间、能量消耗和资源利用率三方面参数来综合评价任务的协同成本。

当机器人 R_i 选择执行任务 T_j 时,所花费的时间用 t_{ij} 来表示,可由式(1)计算求得:

$$t_{ij} = \frac{D(R_i, T_j)}{V_i} = \frac{\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}}{V_i} \quad (1)$$

其中, $D(R_i, T_j)$ 表示机器人位置 (x_i, y_i) 与任务位置 (x_j, y_j) 之间的欧氏距离, V_i 表示机器人的移动速度。

任务的完成时间与协同小组中机器人的到达时间有关,假设任务 T_j 所分配的小组为 G_j 。 T_j 的完成时间 τ_j 计算见式(2):

$$\tau_j = \max t_{ij}, i \in G_j \quad (2)$$

$$\text{Time}_j = 1 - e^{-\tau_j} \quad (3)$$

本文把 G_j 执行任务 T_j 所花费的时间成本用 Time_j 参数表示,可知 T_j 的完成时间越早,该参数越小。

机器人在移动过程中会消耗能量,移动距离越远,消耗的能量越多。将 G_j 执行任务 T_j 所花费的能量消耗用 Ener_j 参数表示,计算公式如下:

$$\text{Ener}_j = 1 - e^{-\frac{\sum_{i=1}^N \varepsilon D(R_i, T_j)}{\sqrt{(l_{\text{width}} + l_{\text{length}})^2 + C_1}}} \quad (4)$$

其中, N 表示 G_j 中机器人的个数; ε 表示能量消耗参数; C_1 表示一个大于零的常数; l_{length} 和 l_{width} 分别表示任务空间的总长度和总宽度。

为了更好地评估 G_j 中机器人资源的使用情况,使用 Rate_j 参数来计算:

$$r(i) = \frac{r_j(i)}{\sum_{s=1}^N c_s(i)} \quad (5)$$

$$\text{rat}(i) = \frac{1}{1 + e^{r(i)}} \quad (6)$$

$$\text{Rate}_j = \prod_{i=1}^m \text{rat}(i) \times \sqrt{\frac{1}{N} \sum_{i=1}^m (\text{rat}(i) - \overline{\text{rat}(i)})^2} \quad (7)$$

其中,式(5)表示 G_j 中 i 类型资源的利用率, N 表示 G_j 中机器人的个数。式(7)通过单类型资源的利用率和多种类型资源利用率之间的标准差来衡量整体资源分配的好坏,目的是使每一种类型资源都有一个较高的利用率且相互之间的差距不大,这里 $\overline{\text{rat}(i)}$ 为每种类型资源利用率的均值。资源整体利用率越高, Rate_j 参数越小。

因此,任务的综合协同成本定义为 U_j ,用于将不同的参数转换为统一的单位:

$$U_j = \alpha \times \text{Time}_j \times \text{Ener}_j \times \text{Rate}_j \quad (8)$$

其中, α 表示转换系数。

本文的优化目标是在资源有限的情况下最小化系统任务的总协同成本,从而使更多的任务能够被成功执行。 y_i 表示任务 T_j 是否能被完成,如果能, $y_i = 1$,否则 $y_i = 0$ 。接下来,研究推得的公式为:

$$\text{Min } U = \sum_{j=1}^k U_j y_j \quad (9)$$

$$\text{s.t: } \sum_{i=1}^N c_i(q) \geq r_j(q), \forall q \in \{1, \dots, m\} \quad (10)$$

$$\tau_j \leq de_j \quad (11)$$

$$\sum_{j=1}^k x_{ij} \leq 1, \forall i \in \{1, \dots, n\} \quad (12)$$

$$x_{ij} \in \{0, 1\}, i \in \{1, \dots, n\}, j \in \{1, \dots, k\} \quad (13)$$

其中,约束(10)表示小组中每种类型资源都应符合任务的资源需求;约束(11)表示任务的执行时间要在其截止时间前;约束(12)和(13)表示每个机器人对于同一个任务最多只能执行一次。

2 基于纳什均衡策略的任务分配方法

为了使任务协同成本最小,本文提出的基于纳什均衡策略的任务分配方法主要分2个阶段:由于任务和机器人数量较多,首先通过遗传算法选出满足任务需求(资源和截止时间)的初始机器人小组,以最大化任务的执行成功率;再针对选择的初始小组,使用博弈论中纳什均衡的思想,通过将各小组的机器人进行互换任务的方法以进一步优化任务的协同成本,最终得到一个分配最优解。任务分配的流程如图2所示。

2.1 基于遗传算法的任务初始协同小组分配

本文采用遗传算法构建任务的初始协同小组。遗传算法是一种运行速度快且应用性较强的启发式搜索算法,已广泛应用在解决最优化问题上,算法的思路是建立在达尔文生物进化论中适者生存和优胜劣汰的原理上,将生物学中的基因编码、染色体交

叉、基因变异以及自然选择等概念引入最优化问题的求解过程中,通过种群的不断选择与淘汰,最终得到问题的最优解。本文算法的具体步骤如下:

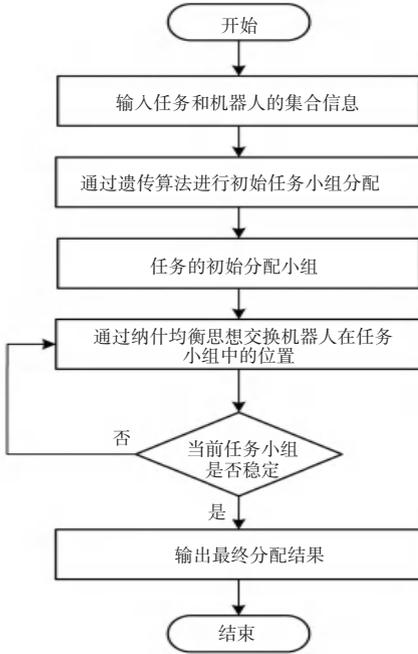


图 2 任务分配流程图

Fig. 2 Flow chart of task assignment

(1)任务分配的编码方式。将任务分配结果用 0-1 矩阵表示,矩阵的行和列分别表示 K 个任务和 N 个机器人,分配矩阵 $x_{ij}(i = 1, 2, \dots, K; j = 1, 2, \dots, N) = 1$ 表示机器人 R_i 分配给任务 T_j , 否则未分配, 则 $G_j = \sum_{i=1}^N x_{ij}$ 表示任务 T_j 的协同小组。该编码方式中每个任务的分配结果作为一个染色体, 每个个体表示所有任务的分配结果。协同小组的分配矩阵如图 3 所示。

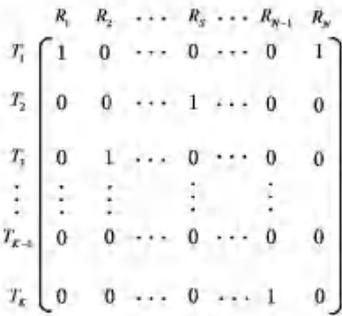


图 3 任务协同小组分配矩阵示例

Fig. 3 Example of task collaboration group assignment matrix

(2)初始化种群。假设种群规模为 pop_N , 即个体 $X_1, X_2, \dots, X_{pop_N}$ 。为了提高解的搜索效率, 本文使用贪婪方法来初始化种群。对于每个 X_i , 从任务

集中随机选择一个任务, 贪婪地从机器人集合中选择与任务距离最小的机器人, 直到满足该任务的需求, 然后再选择其他任务重复此步骤, 直到系统没有可用的资源为止, 并根据分配结果对分配矩阵中的元素进行赋值。在分配时, 要保证机器人只能属于一个任务, 即每一列的和为 1。

(3)选择操作。选择操作是保留适应度高的个体, 淘汰适应度低的个体, 以提高遗传算法的全局搜索能力。本算法以系统任务的执行成功率 $Task_{suc}$ 作为个体分配方案的适应度函数 $Fit(X_i)$, 采用轮盘赌的方法选择具有较好性能的个体, 每个个体被选择的概率与其适应度比值成正比, 适应度越大表示任务执行成功率越高。此处需用到的公式可写为:

$$Task_{suc} = \frac{\sum_{j=1}^n y_j}{n} \quad (14)$$

$$Fit(X_i) = \frac{Task_{suc}(X_i)}{\sum_{k=1}^{pop_N} Task_{suc}(X_k)} \quad (15)$$

其中, y_i 表示任务是否能成功完成, 根据矩阵中每个任务的协同小组 G_j 是否满足式 (10) 和 (11), 如果满足, 则 $y_i = 1$, 否则 $y_i = 0$ 。

(4)交叉操作。从父代中随机选择若干对个体, 每一对根据交叉概率对机器人的分配结果进行交叉操作。本算法采用以矩阵的列为单位进行部分匹配交叉, 随机选择每对个体中要交叉的列, 然后互相交换个体间对应的列以生成 2 个新个体。假设有 7 个机器人和 3 个任务需要进行任务分配, 图 4 表示具体的交叉方式。

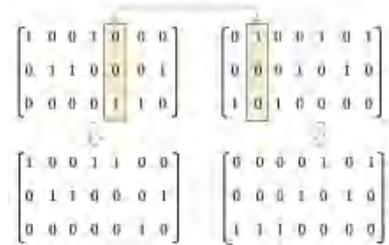


图 4 交叉操作示例图

Fig. 4 Example diagram of cross operation

(5)变异操作。为了避免算法陷入局部最优并加快收敛速度, 随机选择矩阵中的几列, 并在这几列中随机选择某个元素以概率 p_m 改变元素的值, 将元素 '1' 变为 '0' 或者元素 '0' 变为 '1', 同时要保证机器人只能属于任务集中的一个任务, 如图 5 所示, 随机选一行进行变异操作。初始协同小组构建过程

的算法描述如下。



图5 变异操作示例图

Fig. 5 Example diagram of mutation operation

算法1 任务初始协同小组分配算法

输入 任务集合 T , 机器人集合 V , 种群规模 pop_N , 迭代次数阈值 I_N

输出 最优分配矩阵 X_{best}

1. 初始化种群分配结果, 即 $X_1, X_2, \dots, X_{pop_N}$
2. 初始化迭代次数, 令 $Iter = 0$
3. 计算所有个体的适应度值, 根据轮盘赌的方式进行选择操作
4. 对选择的个体进行交叉操作, 以产生新的交叉个体
5. 对选择的个体进行变异操作, 以产生新的变异个体
6. While $Iter < I_N$:
7. 执行步骤 3~5
8. 令 $Iter = Iter + 1$
9. End while
10. 输出最优个体 X_{best}

2.2 基于纳什均衡解的最优协同小组分配

通过上述遗传算法得到的任务协同小组虽然能完成任务, 但是各小组内存在协同成本较高的问题。为了降低组内机器人的协同成本, 本文使用纳什均衡的方法来做进一步优化。纳什均衡是博弈论中一种解的概念, 以美国数学家 (John Forbes Nash) 命名, 是指满足以下性质的策略组合: 任何一位玩家在此策略组合下单方面改变自己的策略 (其他玩家策略不变) 都不会提高自身的收益。本文通过对不同任务小组中的成员进行分配结果相互交换, 使系统的分配策略在之前分配的基础上不断地优化, 最终没有任何一个机器人能通过互换的方式来降低总协同成本, 则系统任务的分配策略满足纳什均衡解。

任务经过初始分配后, 可以得到一个不相交的任务协同小组 $G = \{G_1, G_2, \dots, G_k\}$, $k \leq K$, 且 $G_i \cap G_j = \emptyset (i \neq j)$, 用 G_0 表示未分配任务的机器人集合。对于 G 中每个 G_i 都能成功完成对应的任务 T_j , 此时系统的总协同成本用 U 表示:

$$U = \sum_{i=1}^k U_i \quad (16)$$

$$D_{mn} = U_i + U_j - \tilde{U}_i(r_{mi} \leftrightarrow r_{nj}) - \tilde{U}_j(r_{nj} \leftrightarrow r_{mi}) \quad (17)$$

$$D_{ab} = U_p - \tilde{U}_p(r_{ap} \leftrightarrow r_{b0}) \quad (18)$$

通过不断交换机器人之间的分配结果来寻找纳什均衡解, 目的是降低系统总协同成本 U 。式 (17) 中的 D_{mn} 表示机器人 r_m 与 r_n 相互交换任务分配结果后的成本降低值, 即任务 T_i 小组中的机器人 r_{mi} 和任务 T_j 小组中的机器人 r_{nj} 相互交换任务小组, U_i 和 $\tilde{U}_i(r_{mi} \leftrightarrow r_{nj})$ 分别表示任务 T_i 交换前后的协同成本。式 (18) 中的 D_{ab} 表示未分配集合 G_0 中的机器人 r_{b0} 与任务小组 T_p 中的机器人 r_{ap} 交换后的成本降低值。因此, 只有当 D_{mn} 或 D_{ab} 大于 0 时才是有效的互换, 这一过程的算法描述如下。

算法2 满足纳什均衡解的任务协同小组分配算法

输入 任务初始分配集 G , 未分配机器人集 G_0

输出 纳什均衡解 G_{Nash}

1. 随机选择任务小组 $G_i \in G$
2. 从 G_i 中任选一个机器人 $r_{ki} \in G_i$
3. 初始化 $E_i = \emptyset$
4. 通过式 (17) 计算 r_{ki} 和 G 中其他小组机器人的 D_{kn} 值, 并将 $D_{kn} > 0$ 的结果放在 E_i 集合
5. 通过式 (18) 计算 r_{ki} 和 G_0 中空闲机器人的 D_{km} 值, 并将 $D_{km} > 0$ 的结果放在 E_i 集合中
6. if $E_i \neq \emptyset$:
根据 E_i 集合中的最大值进行相互交换, 并更新 G 和 G_0 集合
7. else:
 G 和 G_0 不做改变
8. 不断重复步骤 1~7, 直到 G 中的信息不再改变为止
9. 输出 G_{Nash}

3 实验分析

3.1 实验设置

在实验环境方面, 本文使用 Python3.6 在 Win10 操作系统进行仿真实验, 实验中的主要参数设置见表 1。本文假设任务共有 3 种类型的资源需求, 即 $m = 3$, 每个机器人携带的资源类型为 3 种类型的随机组合, 并在环境空间中随机生成机器人和任务的位置坐标。

为了评估文本算法的性能, 本文将与其他 2 个算法 (最小能量消耗算法 (MEC) 和基于任务最优完成时间算法 (TOCT) [20]) 作为对比算法, 针对不同的分配场景进行性能对比。在 MEC 中, 根据符合指定

任务资源需求且能量消耗最小的条件,为任务选择机器人小组;在 TOCT 中,根据机器人与任务之间的移动时间来构建博弈论公式,将每个机器人分配到适当的任务小组中,在满足任务资源需求的前提下使任务的完成时间最早。

表 1 参数设置

Table 1 Parameter settings

参数	取值范围
环境空间	100×100
任务数量	[10,150]
机器人数量	[100,300]
任务需求 $r_j(i)$	[2,4]
机器人资源 $c_i(q)$	[1,5]
机器人速度 V_i	10
机器人能量 e_i	50
转换系数 α	100

为了验证所提方法的有效性,本文提出了任务完成率、任务平均协同成本两个性能指标进行评估,分析任务与机器人在不同数量的情况下对各个指标的影响,每种情况运行 10 次实验然后取平均值作为实验结果。

(1)任务完成率。具体计算公式如下:

$$\text{任务完成率} = \frac{\text{已完成的任务数量}}{\text{总任务数量}} \quad (19)$$

本文将任务完成率定义为机器人协同小组成功执行的任务数量与系统发布的总任务数量的比值。

(2)平均协同成本。具体计算公式如下:

$$\text{平均协同成本} = \frac{\text{总协同成本}}{\text{已完成的任务数量}} \quad (20)$$

本文将平均协同成本定义为总协同成本与已完成任务数量的比值,以整体衡量每个任务的执行成本大小,总协同成本为单个任务协同成本 U_j 的累加和。

3.2 实验结果分析

(1)任务完成率。本文采用定量分析法开展仿真实验,图 6 展示了当机器人数量固定为 300 时,任务数量变化对任务完成率的影响。从图 6 中可以看出,随着任务数量的增加,3 种算法的任务完成率都呈现递减的趋势。其原因是随着任务数量的增加,每个任务对有限的机器人资源竞争变大,当任务数量达到 90 时,任务的资源分配逐渐达到饱和状态,后续没有足够的机器人再进行任务分配;同时,本文也研究了当任务数量固定为 100 时,机器人数量变化对任务完成率的影响,如图 7 所示。由图 7 可知,

任务完成率随机器人数量的增加呈现递增的趋势,其原因是当机器人数量较少时,能够符合任务资源需求的机器人也相对较少,从而可完成的任务比例也较低。在 3 个算法中,本文算法始终具有最高的完成率,MEC 算法的效果相对较差。分析可知就是 MEC 算法中的机器人根据能量消耗最小原则选择任务,导致每个任务所分配机器人小组中的资源在类型和数量上与任务需求之间的匹配度较低,进而使较多的任务分配失败。而本文算法和 TOCT 算法相比,由于本文算法初始时使用遗传方法来构建任务小组,以最大限度地找出完成率较高的解方案,而 TOCT 算法只是针对任务完成时间最早原则来选择机器人小组,导致某些任务由于错过了周围携带某种类型资源的机器人而不能被成功执行,因此本文算法的任务完成率较高。

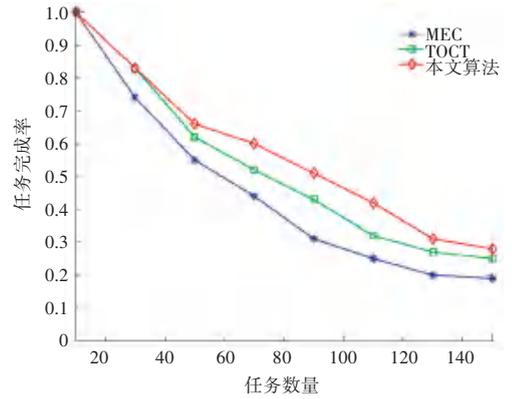


图 6 任务数量对任务完成率的影响

Fig. 6 The effect of task quantity on task completion rate

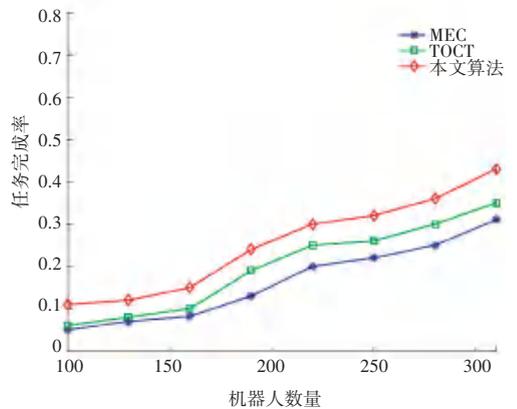


图 7 机器人数量对任务完成率的影响

Fig. 7 The effect of the number of robots on task completion rate

(2)平均协同成本。当机器人数量固定为 300 时,图 8 显示任务数量变化对任务平均协同成本的影响。在任务数量低于 50 时,3 种算法的平均协同成本较高,随着任务数量的增加,其平均协同成本逐渐降低。这是因为任务较少时,机器人资源相对充

足,进而会花费较大的成本去完成一些开始时间较晚且位置相对偏远的任务,随着任务数量的增加,机器人可选择的任务较多,因此会选择一些协同成本相对较低的任务。图9显示了当任务数量固定为100时,机器人数量变化对任务平均协同成本的影响,随着机器人数量的增加,每种算法的平均协同成本逐渐增加。这是因为当机器人数量较少、任务数量较多时,机器人也会偏向执行协同成本较低的任务,但随着机器人数量的增加一些协同成本较大的任务相继被完成,从而使任务的平均成本有所增加。在3个算法中,本文算法的协同成本始终保持最低的水平,这是因为MEC和TOCT在分配任务协同小组时,考虑的都是单一的优化目标,忽略了资源利用率、能量消耗和任务完成时间对任务执行效率的整体影响,导致任务的平均成本较大。而本文算法在任务初始小组创建完后,通过不断地移动机器人在小组中的位置来寻找纳什均衡解,以进一步降低协同成本,因此任务的平均协同成本较低。

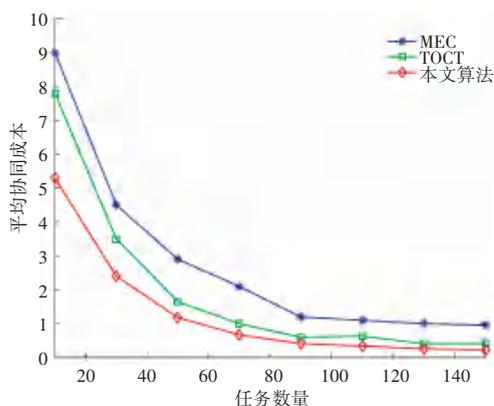


图8 任务数量对任务平均协同成本的影响

Fig. 8 The effect of the number of tasks on the average synergistic cost of a task

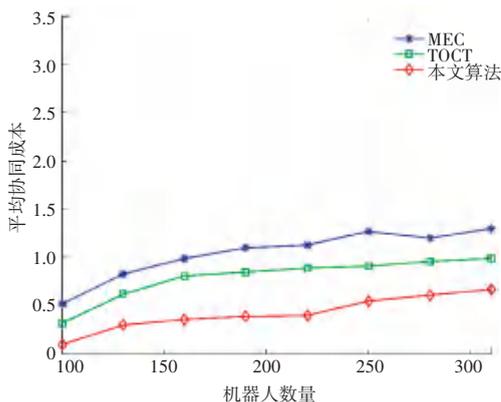


图9 机器人数量对任务平均协同成本的影响

Fig. 9 The effect of the number of robots on the average collaborative cost of a task

4 结束语

本文针对多机器人系统中任务的多类型资源需求问题,提出了面向多类型资源需求的机器人协同小组任务分配方法。为了降低任务的协同成本,该方法考虑了影响任务执行的时间、能量消耗和资源利用率等因素。在任务分配时,本文首先通过遗传算法为任务分配初始的机器人小组,然后利用博弈论中的纳什均衡方法对初始机器人小组做进一步优化。通过仿真实验表明本文提出的方法在提高任务完成率、降低任务执行成本方面具有较好的效果。在未来的工作中,应该更多考虑机器人执行任务时的资源补充和续航等影响任务分配的因素,并探索更高效的智能方法来优化任务的分配过程。

参考文献

- [1] WEINBERG G V. Prediction of UAV swarm defeat with high-power radio frequency fields [J]. IEEE Transactions on Electromagnetic Compatibility, 2022, 64(6): 2157-2162.
- [2] SCHIANO F, KORNATOWSKI P M, CENCETTI L, et al. Reconfigurable drone system for transportation of parcels with variable mass and size [J]. IEEE Robotics and Automation Letters, 2022, 7(4): 12150-12157.
- [3] KORNATOWSKI P M, FEROSKHAN M, STEWART W J, et al. A morphing cargo drone for safe flight in proximity of humans [J]. IEEE Robotics and Automation Letters, 2020, 5(3): 4233-4240.
- [4] KUANG Zhenfei, WEI Wei, YAN Yier, et al. A real-time and robust monocular visual inertial SLAM system based on point and line features for mobile robots of smart cities toward 6G [J]. IEEE Open Journal of the Communications Society, 2022, 3: 1950-1962.
- [5] LIU Huaping, DENG Yuhong, GUO Di, et al. An interactive perception method for warehouse automation in smart cities [J]. IEEE Transactions on Industrial Informatics, 2020, 17(2): 830-838.
- [6] LIU Xilong, ANSARI N. Resource allocation in UAV-assisted M2M communications for disaster rescue [J]. IEEE Wireless Communications Letters, 2018, 8(2): 580-583.
- [7] DONG Jiong, OTA K, DONG Mianxiang. UAV-based real-time survivor detection system in post-disaster search and rescue operations [J]. IEEE Journal on Miniaturization for Air and Space Systems, 2021, 2(4): 209-219.
- [8] DUTTA A, ASAITHAMBI A. One-to-many bipartite matching based coalition formation for multi-robot task allocation [C]//2019 International Conference on Robotics and Automation (ICRA). Montreal, Canada: IEEE, 2019: 2181-2187.
- [9] CHEN Jianping, WANG Jianbin, XIAO Qijun, et al. A multi-robot task allocation method based on multi-objective optimization [C]//2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV). Singapore: IEEE, 2018: 1868-1873.
- [10] ZHU Qinqun, OH J C. Learning fairness under constraints: A decentralized resource allocation game [C]//2016 15th IEEE

- International Conference on Machine Learning and Applications (ICMLA). Anaheim, USA; IEEE, 2016; 214–221.
- [11] ZITOUNI F, HAROUS S, MAAMRI R. A distributed approach to the multi-robot task allocation problem using the consensus-based bundle algorithm and ant colony system[J]. IEEE Access, 2020, 8; 27479–27494.
- [12] HUANG Liwei, QU Hong, ZUO Lin. Multi-type UAVs cooperative task allocation under resource constraints[J]. IEEE Access, 2018, 6; 17841–17850.
- [13] WU Danfeng, ZENG Guangping, MENG Lingguo, et al. Gini coefficient-based task allocation for multi-robot systems with limited energy resources[J]. IEEE/CAA Journal of Automatica Sinica, 2017, 5(1): 155–168.
- [14] NOTOMISTA G, MAYYA S, EMAM Y, et al. A resilient and energy-aware task allocation framework for heterogeneous multirobot systems[J]. IEEE Transactions on Robotics, 2021, 38(1): 159–179.
- [15] CHEN Xinye, ZHANG Ping, DU Guanglong, et al. A distributed method for dynamic multi-robot task allocation problems with critical time constraints[J]. Robotics and Autonomous Systems, 2019, 118; 31–46.
- [16] TURNER J, MENG Qinggang, SCHAEFER G, et al. Distributed task rescheduling with time constraints for the optimization of total task allocations in a multirobot system[J]. IEEE Transactions on Cybernetics, 2017, 48(9): 2583–2597.
- [17] CZATNECKI E, DUTTA A. Hedonic coalition formation for task allocation with heterogeneous robots [C]//2019 IEEE International Conference on Systems, Man and Cybernetics (SMC). Bari, Italy; IEEE, 2019; 1024–1029.
- [18] BISCHOFF E, MEYER F, INGA J, et al. Multi-robot task allocation and scheduling considering cooperative tasks and precedence constraints[C]//2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC). Toronto, Canada; IEEE, 2020; 3949–3956.
- [19] 熊乾程,董晨,洪祺瑜. 面向智能工厂原料供应环节的多机器人任务分配方法[J]. 小型微型计算机系统, 2022, 43(8): 1625–1630.
- [20] LI Qinyuan, LI Minyi, VO B Q, et al. Distributed near-optimal multi-robots coordination in heterogeneous task allocation[C]//2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Las Vegas, USA; IEEE, 2020; 4309–4314.