

文章编号:1671-6833(2018)02-0006-05

一种基于改进适应度的多机器人协作策略

曾庆山,冯珊珊

(郑州大学 电气工程学院,河南 郑州 450001)

摘 要:通过对基于适应度的协作策略及其改进方法的研究发现,针对机器人在两个任务的适应度相同时无法选择出最匹配任务的问题,提出通过加入与机器人起止位置有关的距离适应度函数,使得机器人可以选择出最优匹配任务;同时,针对外部能力适应度,采用更符合实际的高斯分布模型来计算适应度.仿真结果表明,改进后的算法不仅实现了最优匹配,而且算法更高效,更节省能量.

关键词:适应度;协作策略;多机器人;高斯分布;距离适应度

中图分类号:TP242 文献标志码:A doi:10.13705/j.issn.1671-6833.2018.02.003

0 引言

多机器人协作策略是多机器人系统执行任务的前提,也是系统执行效率是否高效的关键,是组织简单机器人完成复杂任务的核心.尤其近几年云计算的兴起,任务分配的相关算法在云调度<sup>[1]</sup>中也发挥了重要的作用,因此具有重要的研究价值.

目前,多机器人协作策略研究主要集中于3种方法<sup>[2]</sup>:市场机制、群体智能和建模法.在动态环境中,多机器人系统经常面临任务在不确定的时间和不断变化的位置发布的问题<sup>[3]</sup>.建模法最主要的优点就是实时性强,尤其在这种动态情况下,能较快地针对环境做出合理高效的决策.董炆斌等<sup>[4]</sup>创建的基于适应度的多机器人协作策略,在针对动态松散多任务分配方面展现了较好的实时性和灵活性,且能够在限制任务执行优先级、机器人能力有限<sup>[5]</sup>等情况下完成较优的任务分配,仿真试验表明,算法也具有较好的稳定性.冯晓海等<sup>[6]</sup>通过对该算法的改进,使用正余切函数作为外部适应度函数,避免了原始函数中一直归一化计算的问题,提高了机器人与任务之间适应度计算的稳定性.

1 适应度协作策略及其不足

1.1 外部适应度算法

外部能力适应度的计算如下:

$$A_{ij}(t) = \begin{cases} \sum_{k \in U_n} W_{j,k} \times L_{i,k}, & \text{如果 } U_n \subseteq U_p \\ 0, & \text{否则} \end{cases} \quad (1)$$

外部能力适应度的归一化处理方法,

$$\bar{A}_{ij}(t) = A_{ij}(t) / \max A_{ij}(k), 0 \leq \bar{A}_{ij}(t) \leq 1. \quad (2)$$

该算法很好地建立了任务的内部状态和外部联系,在机器人选择任务过程中,任务的内部状态即内部适应度保持不变,不会因为机器人的不同而改变.这也说明,机器人选择任务的关键在于它与任务的外部适应度,外部适应度的设置是否合理将直接影响到任务分配的效果.显然,外部适应度过度依赖全局信息且频繁进行归一化的过程增加了系统的复杂度,降低了稳定性.

1.2 正余切外部适应度算法

正余切外部适应度如下:

$$P(t) = \begin{cases} \frac{2}{\pi} \operatorname{arccot} \left( \frac{E_r}{E_t} - 1 \right), & (E_t \leq E_r) \\ \frac{2}{\pi} \arctan \left( \frac{E_r}{E_t} - 1 \right), & (E_t > E_r) \end{cases}, \quad (3)$$

式中:  $E_r$  为机器人的实际能力;  $E_t$  为任务的实际需求能力.

改进的方法提高了算法的独立性,但也存在一些不合理:改进的外部适应度在较符合能力要求的范围内适应度值下降较快,在远超过能力的范围内适应度值保持较大值且下降缓慢.在计算适应度前,系统是有开关控制变量的,当机器人能力不足时,不进行适应度计算,

收稿日期:2017-11-02;修订日期:2017-12-29

作者简介:曾庆山(1963—),男,湖北武汉人,郑州大学教授,博士,主要研究方向为智能控制理论、复杂系统的建模与控制、分数阶微分控制等,E-mail:qszeng@126.com.

它必须与其他机器人的能力进行加和、并操作后才能进行适应度的计算<sup>[7]</sup>.在进行加和、并操作时,把能力不足时的适应度作为联合的依据不能保证联合的机器人组合与任务最优匹配,因此,对机器人能力不足时的外部适应度计算是多余的.

## 2 改进的外部适应度

### 2.1 改进外部能力适应度

在机器人能力满足任务需求时,计算能力适应度时,一般要求在较符合能力要求的范围内,能力适应度较大且区分明显;在远超出能力要求的范围内,能力适应度较小.为实现这一要求,笔者设计了基于高斯分布模型的外部能力适应度函数,如下式:

$$N_{ij} = \exp\left(\frac{(L(i) - W(j))^2}{2 \times W(j)^2}\right), L(i) \geq W(j), \quad (4)$$

式中:  $L(i)$  为机器人  $R_i$  的实际能力水平;  $W(j)$  为任务  $T_j$  的实际能力需求. 其仿真对比如图 1、2.

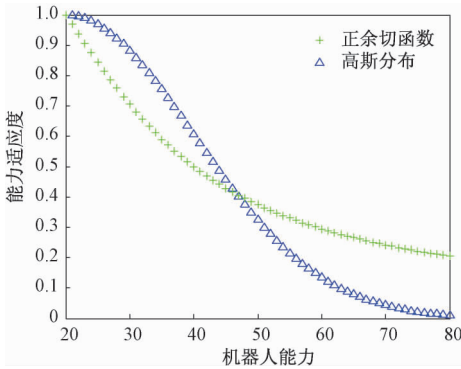


图 1  $W=20$  时的适应度值对比

Fig.1 Fitness comparison when  $W=20$

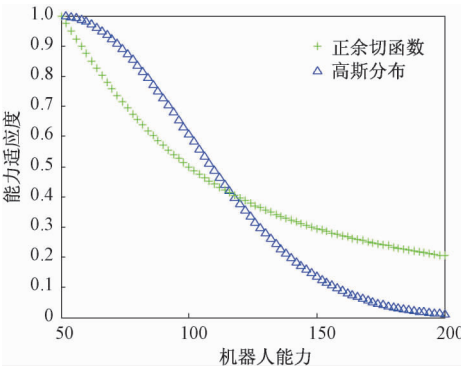


图 2  $W=50$  时的适应度值对比

Fig.2 Fitness comparison when  $W=50$

从图 1、2 可以看出,在任务的能力需求是 20(50)时,机器人在 40(100)以内的适应度要

比反余切函数的大,超过 40(100)以后,适应度迅速减小,比反余切函数要小得多,可以保证其它能力在任务选择中占据主导,选择任务的匹配度更高.算法的计算只与该任务的能力需求和当前机器人的能力有关,不受系统里其它因素的影响,确保算法的独立性、稳定性和可延拓性.

### 2.2 距离适应度策略

图 3 为机器人不合理的任务选择.如图 3 所示,机器人起止位置异地,当 A 选择任务 1 后,B 对 1、2 两个任务的适应度相同,按照原适应度算法的选择原则,由于 1 任务的编号在前,则机器人 B 将选择任务 1,导致任务 2 漏选,从位置上看机器人 B 选择任务 2 更合理.显然,外部适应度如果只包含能力的匹配而不包含位置的匹配是不合理的,由此考虑加入机器人的代价<sup>[8]</sup>.

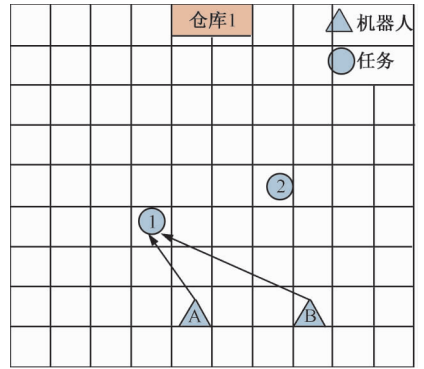


图 3 机器人不合理的任务选择

Fig.3 The unreasonable task selection of robots

基于此,笔者在外部适应度的结构中加入了与机器人起止位置有关的距离适应度函数,

$$Fdis_{ij} = \begin{cases} \frac{Ldis - RTdis(ij)}{Ldis}, & \text{机器人起止异地} \\ \frac{RTdis(ij)}{Ldis}, & \text{机器人起止同地} \end{cases}, \quad (5)$$

式中:  $Fdis$  为距离适应度;  $Ldis$  为任务空间的最大距离,一般在任务之前就已经确定;  $RTdis(ij)$  为机器人  $R_i$  与任务  $T_j$  之间的距离.

外部适应度函数改为:

$$A_{ij} = W_n \cdot N_{ij} + W_f \cdot Fdis_{ij}, \quad (6)$$

式中:  $W_n + W_f = 1, 0 < W_n < 1, 0 < W_f < 1$ .

任务分配中,不仅涉及到任务执行的时间效率,还要考虑机器人的能量消耗问题,只有这样才能保证任务是最优分配.为方便计算,给出多机器人执行任务的能量消耗计算公式,

$$E(i) = E_i \cdot \sum_{j=1}^n (LT_{ij}), n = 1, 2, 3, \dots, T_n, \quad (7)$$
$$E_i = L_i/2, \quad (8)$$

式中:  $E(i)$  为机器人  $R_i$  执行完所有任务的能量消耗;  $E_i$  为机器人  $R_i$  每行走 1 m 消耗的能量, 与机器人的能力水平  $L_i$  相关, 机器人能力水平越大, 则单位消耗的能量越大;  $LT_{ij}$  为机器人  $R_i$  执行任务  $T_j$  的行走路程。

改进后的算法与之前的算法相比有如下优点。

①改进后, 针对图 3 中的问题, 由于加入了距离适应度, 使得机器人 B 对任务 2 的适应度更大, 从而选择出了更合理且更优的任务。

②机器人起止位置异地. 当机器人对任务的其他适应度相同时, 机器人距离任务越近, 则适应度越大. 这样, 机器人可以缩短一次返回行走的距离, 节约时间. 尤其是在大规模的任务分配中, 将大大减少执行时间, 如图 4 所示. 图 4 中机器人 A 要将任务 1、2 送到仓库 1. 改进前 A 的任务执行顺序为:  $A \rightarrow 1 \rightarrow \text{仓库 1} \rightarrow 2 \rightarrow \text{仓库 1}$ , 则机器人 A 的行程为 21.59 m; 改进后 A 的执行任务顺序为:  $A \rightarrow 2 \rightarrow \text{仓库 1} \rightarrow 1 \rightarrow \text{仓库 1}$ , 机器人 A 的行程为 18.10 m, 缩短了行程。

③机器人起止位置相同. 当机器人对任务的其他适应度相等时, 距离越远的适应度越大. 我们一般认为, 当机器人与任务能力最匹配时达到资源的充分利用. 因此, 让最优匹配的机器人选择更远距离的任务, 让过匹配的机器人去执行较近的, 这样, 在总行走距离相等的情况下, 总体节约了能量. 如图 4, 让机器人 B、C 分别将任务 1、2 送到仓库 1. 其中, B 对两任务的适应度相同且属于最优匹配, C 对两任务是属于过匹配. 改进前分配结果为:  $B \rightarrow 1, C \rightarrow 2$ ; 改进后分配结果为:  $B \rightarrow 2, C \rightarrow 1$ . 根据公式(7)计算可知改进后更节省能量, 证明如下。

已知:  $0 < L(B) < L(C), 0 < LT_1 < LT_2, LT_1、LT_2$  代表仓库 1 分别到任务 1、2 的距离。

改进前的能量消耗:

$$E(B) = L(B)/2 \cdot LT_1, E(C) = L(C)/2 \cdot LT_2,$$
$$E1 = L(B)/2 \cdot LT_1 + L(C)/2 \cdot LT_2.$$

改进后的能量消耗:

$$E(B) = L(B)/2 \cdot LT_2, E(C) = L(C)/2 \cdot LT_1,$$
$$E2 = L(B)/2 \cdot LT_2 + L(C)/2 \cdot LT_1.$$
$$\Delta E = E1 - E2$$
$$= L(B)/2 \cdot LT_1 + L(C)/2 \cdot LT_2 -$$

$$(L(B)/2 \cdot LT_2 + L(C)/2 \cdot LT_1) =$$
$$(L(B) - L(C))(LT_1 - LT_2)/2 > 0.$$

证毕.

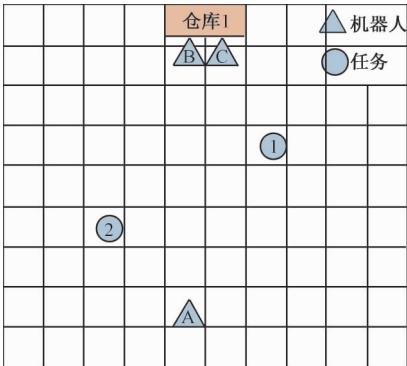


图 4 机器人和任务分布图

Fig.4 The distribution of robots and tasks

3 仿真试验

以搬运任务进行验证, 在一个  $10 \times 10$  m 的空间内, 配备有 4 个机器人, 为进行对比仿真验证, 每个机器人的能力水平各不相同, 机器人起始位置也不相同, 任务的能力需求、位置、数量也各不相同, 在执行任务期间, 机器人均以 0.24 m/s 的速度行进, 仿真试验分别对不同组合的任务进行分配和结果分析。

表 1 是机器人的初始参数, 其中,  $R_i$  是机器人编号;  $RPosX、RPosY$  是机器人坐标;  $L$  是机器人能力水平;  $CKPos$  是仓库位置. 表 2 是任务的一些初始参数,  $T_j$  是任务编号;  $P$  是任务优先级;  $I$  是子任务总量;  $I_c$  是子任务完成量;  $A_r$  是当前选择本子任务的机器人数量;  $L_{up}$  是允许机器人上限<sup>[9]</sup>;  $L_{dn}$  是允许机器人下限;  $W$  是任务能力需求;  $TPos$  是任务坐标. 初始时, 起止位置异地. 内部适应度参数设置为  $W_s = 0, W_p = 0.8, W_r = 0.2$ , 第一次任务选择时, 要保证机器人尽量选择较近的任务, 参数设置为:  $W_n = 0.2, W_f = 0.8$ , 接下来的参数设置为  $W_n = 0.5, W_f = 0.5$ 。

表 1 机器人初始参数

Tab.1 The initial parameters of robots

$R_i$	$RPosX$	$RPosY$	$L$	$CKPos$
1	3	1	30	5,10
2	5	1	20	5,10
3	7	1	40	5,10
4	9	1	20	5,10

表 2 任务初始参数

Tab.2 The initial parameters of tasks								
$T_j$	$P$	$I$	$I_c$	$A_c$	$L_{up}$	$L_{dn}$	$W$	$TPos$
1	1	3	0	0	4	1	10	3,5
2	1	5	0	0	6	1	20	5,8
3	1	4	0	0	5	1	30	9,3
4	1	4	0	0	5	1	30	7,5
5	1	3	0	0	3	1	10	8,3
6	1	3	0	0	2	1	20	4,6

针对此次试验,我们假设不考虑机器人之间的避碰耗时,机器人在执行任务中能力保持不变,机器人依次进行任务选择.为了保证试验的可靠性和严谨性,笔者特意设计了没有适应度相同的情况和有两个任务适应度相同位置不同的情况,这两个任务分别是任务3和4.试验分别以任务组合(1、2、3),(1、2、3、4),(1、2、3、4、5),(1、2、3、4、5、6)进行4组仿真试验.为了验证试验的有效性,试验还分别用原始适应度算法、正余切改进的适应度算法进行了仿真试验验证,试验结果如表3.

表 3 执行任务时间对比

Tab.3 The contrast of task execution time				
$R_{num}=4$	$T=3$	$T=4$	$T=5$	$T=6$
原算法	111	179	180	205
正余切	100	180	185	201
本算法	99	168	168	180

从表3可以看出, $T=3$ 时,任务中没有适应度相同的任务,原始适应度算法比正余切改进的算法、笔者改进的算法运行时间都要长;当 $T=4\sim6$ 时,任务中有一组会出现刚开始适应度相同的情况,此时笔者改进的算法运行时间明显小于原始算法和正余切改进的算法,说明改进的算法有效且效率更高.

根据仿真试验,可以得到每个机器人的行走距离及能力大小,然后通过公式(7)可以计算出所有任务执行结束后每个机器人的总能量消耗,3种方法的统计结果如表4~6所示.

由以上3组的能量计算可看出,笔者改进的算法能量消耗最低,即使在 $T=3$ 时,正余切改进的算法和笔者算法执行时间相差无几,但能量消耗却大于笔者改进的算法,由此证明笔者改进的算法也实现了能量的消耗最低.

表 4 原算法的能量消耗

Tab.4 The energy consumption of the original algorithm					
$R_i$	$E_i$	$R_i$ 的行走距离			
		$T=3$	$T=4$	$T=5$	$T=6$
1	15	30.34	46.06	46.20	65.00
2	10	27.50	27.46	57.16	57.42
3	20	26.92	57.08	57.18	57.14
4	10	35.24	35.24	50.82	52.90
总能量 E1		1 621	2 460	2 916	3 221

表 5 正余切算法的能量消耗

Tab.5 The energy consumption of tanagent cotangent algorithm					
$R_i$	$E_i$	$R_i$ 的行走距离			
		$T=3$	$T=4$	$T=5$	$T=6$
1	15	30.16	46.08	46.08	56.70
2	10	31.56	31.48	49.24	54.72
3	20	26.86	57.10	57.12	57.06
4	10	31.32	31.28	58.70	64.06
总能量 E2		1 618	2 461	2 913	3 180

表 6 笔者改进算法的能量消耗

Tab.6 The energy consumption of the improved algorithm					
$R_i$	$E_i$	$R_i$ 的行走距离			
		$T=3$	$T=4$	$T=5$	$T=6$
1	15	30.18	48.10	48.12	57.16
2	10	25.52	25.50	49.16	56.76
3	20	26.88	53.28	53.30	57.16
4	10	31.28	31.26	49.78	54.50
总能量 E3		1 558	2 355	2 777	3 113

4 结论

主要研究了基于改进适应度的多机器人协作策略,针对适应度算法的外部能力匹配度不够理想的问题,提出了高斯分布模型来计算机器人的能力匹配度.仿真表明,笔者算法更符合匹配度的要求,且保证了算法的独立性与延拓性.针对出现适应度相同的任务时,机器人可能无法选择出最合理任务的问题,通过加入与机器人起止位置有关的距离适应度,不仅解决了适应度相同时依次分配带来的不合理,提高了算法分配合理性,而且任务的执行效率也明显提升,保证能量消耗最低,在松散多机器人环境中,该算法优于改进前的算法.

参考文献：

[1] 王俊英,颜芬芬. 基于概率自适应蚁群算法的云任务调度方法[J]. 郑州大学学报(工学版),2017,38(4):51-56.

[2] 张崧,刘淑华. 多机器人任务分配的研究进展[J]. 智能系统学报,2008,3(2):115-120.

[3] CHEN Y, MAO X J, HOU F, et al. Combining re-allocating and re-scheduling for dynamic multi-robot task allocation [C] // IEEE International Conference on Systems, Man, and Cybernetics. Budapest:IEEE,2016:000395-000400.

[4] 董炆斌,蒋静坪. 基于适应度的多机器人任务分配策略[J]. 浙江大学学报(工学版),2007,41(2):272-277.

[5] MITICHE H, BOUGHACI D, GINI M. Efficient heuristics for a time-extended multi-robot task allocation problem [C] // International Conference on New Technologies of Information & Communication, 2015:1-6.

[6] 张万绪,冯晓海,赵江波,等. 改进适应度的异构多机器人任务分配[J]. 西北工业大学(自然科学版),2013,43(1):22-26.

[7] 柳林,季秀才,郑志强. 基于市场法及能力分类的多机器人任务分配方法[J]. 机器人,2006,28(3):337-343.

[8] 韦兆文,区云鹏,闫俊燕. 一种改进的动态合同网协议[J]. 计算机工程与应用,2007,43(36):208-211.

[9] LERMAN K, GALSTYAN A. A general methodology for mathematical analysis of multi-agent systems [R]. Los Angeles:USC information sciences technical report ISI-TR-529,2001.

A Multi-robot Cooperative Strategy Based on Improved Fitness Function

ZENG Qingshan, FENG Shanshan

(School of Electrical Engineering, Zhengzhou University, Zhengzhou 450001, China)

**Abstract:** This study focused on the fitness-based cooperative strategy and its improvement method. It was found that the problem of the most matching task could not be decided when the robot had the same fitness tasks. By adding the distance fitness function related to the robot’s starting and ending position, the robot could choose the best matching task. At the same time, more realistic Gaussian distribution model was used to calculate the fitness of the external ability. The simulation results showed that the improved algorithm could not only achieve the optimal matching, but also be more efficient and energy saving than before.

**Key words:** fitness; cooperative strategy; multi-robot; heterogeneity; distance fitness