

两段探测目标的传感器任务调度问题 0-1 规划模型及算法*

李建平¹, 张 晗¹, 罗 永¹, 朱 承², 何文涛²

(1. 国防科技大学 理学院, 湖南 长沙 410073; 2. 国防科技大学 信息系统工程重点实验室, 湖南 长沙 410073)

摘要:为解决指挥系统控制中的调度困难,研究了一类特殊的传感器资源调度问题。主要分析了跟踪目标的探测次数、时间间隔和传感器资源等约束条件。用跟踪目标的重要程度之和作为目标函数,建立了一个 0-1 规划的数学模型,再利用变换将其转化为 0-1 线性整数规划模型。利用割平面法求解得出最优调度策略,其能在工作量饱和的情况下合理调度传感器资源。为提高求解速度,提出了对应的模拟退火算法。通过对一些不同规模实例的求解,在资源利用率和算法的求解速度等指标上,与割平面法及遗传算法进行对比分析,验证了模型的有效性和模拟退火算法求解的高效性。

关键词:传感器;任务调度;0-1 规划;模拟退火算法;遗传算法

中国分类号: O221. 4 **文献标志码:** A **文章编号:** 1001-2486(2017)03-121-09

The 0-1 programming model and algorithm for the problem of sensor task scheduling for double detection

LI Jianping¹, ZHANG Han¹, LUO Yong¹, ZHU Cheng², HE Wentao²

(1. College of Science, National University of Defense Technology, Changsha 410073, China;

2. The Key Laboratory of Information System Engineering, National University of Defense Technology, Changsha 410073, China)

Abstract: In order to resolve the scheduling difficulty in the control of command system, the resource scheduling problem of a special sensor was studied and the constraint conditions including detected times, the interval between two detections, and the resource restrict of sensor were analyzed. A 0-1 programming model was established and transformed to a 0-1 liner integer model whose objective function is the sum of the importance degree of tracking targets. The optimal solution which can reasonably schedule sensor resource when the workload is saturated was obtained by using the cutting plane algorithm. A corresponding simulated annealing algorithm was proposed to improve the speed of solving and was used to solve some examples. Compared with the cutting plane algorithm and the genetic algorithm in terms of resource utilization and the speed of solving, the validity of the proposed model and the high efficiency of the simulated annealing algorithm were proved.

Key words: sensor; task scheduling; 0-1 programming; simulated annealing algorithm; genetic algorithm

传感器的任务调度是根据一定优化准则,在确定的时间区间内,为传感器网络中的传感器资源安排跟踪任务,以满足对多个移动目标的跟踪要求,从而达到某项或某些指标最优。传感器资源调度能够有效地提高资源利用效率。在军事问题中,合理的传感器调度,例如雷达,可以提高军方的防御能力。在研究的问题中,传感器对目标进行跟踪时需要先后进行两段探测,每段都有最短时长,若一个目标两段探测由不同的传感器完成,则两段探测之间有最短时间间隔。不同移动方向的目标有不同的严重程度,由于传感器资源的限制,当目标过多时,必须有选择地跟踪重要目标,由此产生了这种两段探测目标的传感器资源

调度问题。

目前,许多传感器资源调度问题主要是研究其启发式调度策略。文献[1-3]对基于工作方式优先级的驻留调度算法进行了改进,文献[4]首先提出了驻留时间窗的概念,其中时间窗^[5]是指雷达的实际执行时间在期望发射时间前后能移动的有效范围。驻留时间窗这一概念的提出增大了雷达任务调度的灵活性,原本冲突的任务有了被重新调度的可能。脉冲交错技术^[6]的使用可以提高雷达的时间利用率。Shih 等先后提出了最小空闲模版优先^[7]和基于模版的综合周期^[8]调度算法,这两种算法均为离线模版算法。Gopalakrishnan 等还提出了在线设计模版^[9]的想

* 收稿日期:2016-05-05

基金项目:国家自然科学基金资助项目(61273322)

作者简介:李建平(1965—),男,湖南涟源人,教授,博士,硕士生导师,Email:jpli_1@163.com

法。自适应调度算法^[10]是近年的研究热门。然而,对于两段探测目标的传感器资源调度问题仍然研究较少,目前文献已有的研究与实际问题应用仍有一定的差距。文献[11]提出了旋转情况下波束驻留任务的模型,并给出了一种旋转相控阵雷达的任务调度启发式算法。遗传算法在求解调度问题方面有广泛的应用。文献[12]给出了遗传算法对车间调度问题的求解方法;文献[13]提出一种新的改进自适应遗传算法,能够在优化过程中自动给出比较合适的交叉概率和变异概率,使算法在保持群体多样性的同时提高了速度;文献[14]等提出一种双层子代产生模式的改进遗传算法应用于车间调度问题,以使子代更好地继承父代的优良特征。

资源分配问题大多数归结为一个整数规划模型的求解。基于此事实,本文选用 0-1 变量对传感器调度问题进行建模和分析。

1 问题描述

当传感器探测目标时,每一个传感器都具有一定的探测范围,当该传感器的探测范围内有目标出现时,传感器可在极短时间内计算出目标移动的轨迹。即当目标进入探测区域内其任意时刻的位置视作已知。朝不同方向移动的目标有不同的目标重要程度,这是由目标本身决定的,传感器必须有选择地跟踪重要程度大的目标。

在传感器资源调度问题中,成功跟踪一个目标是指该目标在飞行过程中先后两次被(同一个或不同的)传感器探测到,并且每次探测的持续时间必须大于一个给定的时间,称这个给定的时间为最短探测时间。同时对某一目标的先后两次探测时间也需要有一定的时间间隔。两段探测可以由不同的传感器进行,但一段探测只能由一个传感器进行。若只探测一段或某段探测时间长度不够,则视为跟踪失败。虽然传感器可以同时探测多个目标,但其存在最大工作容量和探测范围的限制。最大工作容量是指传感器能同时探测的最大目标数量。

传感器资源调度问题的目标是寻找合适的调度策略使跟踪成功的目标重要程度之和最大。在这一问题中,需要决策的变量分别是对目标进行探测的传感器编号和两段探测开始的时刻。

2 问题分析与建模

2.1 模型基本假设

在工作饱和情况下,即目标数量超过最大工

作容量时,需要对多个目标有选择地进行跟踪。为避免重复,统一说明本文第 2 节中出现的 i 的取值范围是 $\{1, 2, \dots, n\}$, j 的取值范围是 $\{1, 2, \dots, m\}$, t 的取值范围是 $\{1, 2, \dots, N\}$ 以及上标 k 的取值范围是 $\{1, 2\}$ 。 n 表示任务期间出现的目标的总个数, w_i 表示目标 i 的重要程度。在任务期间传感器位置和探测范围是不变的,用 m 表示传感器总个数, C_j 表示传感器 j 的工作容量, N 表示时间段划分个数。

对于目标 i ,需要传感器探测的两段时间区间分别为 $s_i^1 = (t_i^1, t_i^1 + \Delta t_i^1)$, $s_i^2 = (t_i^2, t_i^2 + \Delta t_i^2)$,对应的传感器编号记作 R_i^1 和 R_i^2 。目标可能在任意时刻进入或离开传感器的探测区域,在模型中引入进入-离开时刻矩阵来描述目标进入和离开对应传感器探测范围的时间。两段探测目标的传感器调度的目标是确定对于每个目标 i 的探测起始时间 t_i^1, t_i^2 以及对应的探测传感器编号 R_i^1 和 R_i^2 ,使得被成功跟踪的目标重要程度之和最大,并满足下述约束条件:

- 1) 对目标 i 的两段探测时长 Δt_i^1 和 Δt_i^2 不小于给定的时长,在饱和和工作情况下探测时长为最短时长;
- 2) 对目标 i 的第一段探测结束后经过一定时间间隔才能开始第二段探测,即 $t_i^1 + \Delta t_i^1 + \Delta t_i < t_i^2$;
- 3) 对目标 i 的第二段探测结束时刻不晚于指定时刻 Te_i^2 ,即 $t_i^2 + \Delta t_i^2 < Te_i^2$;
- 4) 对于任意时刻 t ,传感器 j 同时探测的目标数量不超过其最大工作容量 C_j 。

引入 0-1 变量 X_{jt}^k 作为决策变量,其中 $X_{jt}^k = 1$ 表示从 t 时刻开始传感器 j 对目标 i 进行第 k 段探测($k = 1, 2$), $X_{jt}^k = 0$ 表示从 t 时刻开始传感器 j 对目标 i 没有进行第 k 段探测。

2.2 目标函数分析

为方便起见,引入一个效用值变量 u_i ,表示目标 i 是否两段都被探测。若目标 i 的两段探测都被完成,则 u_i 为 1,否则为 0。这样,目标函数为各 u_i 的重要程度的加权和,即 $w^T u$ 。模型中,目标函数为成功跟踪的目标的重要程度之和,即跟踪效用值的重要程度加权和 $z = w^T u$,其中 $w = (w_1, w_2, \dots, w_n)^T$, $u = (u_1, u_2, \dots, u_n)^T$ 。

2.3 约束条件分析

在工作饱和情况下,对上述约束条件 1~4 进行进一步分析,得到如下 4 个约束条件。

2.3.1 约束条件 1: 两段探测的限制

由决策变量 X_{jt}^k 的含义可知,对于固定的 i 和

k, X_{ij}^k 等于1的个数即为目标 i 第 k 段被探测的次数,记为 u_i^k ,即:

$$u_i^k = \sum_{j=1}^m \sum_{t=1}^N X_{ij}^k, \quad i=1,2,\dots,n; k=1,2 \quad (1)$$

在工作饱和情况下,只对目标的一段进行探测或者对某一段探测多于一次都是对资源的浪费。即对于目标 i ,应满足 u_i^1 和 u_i^2 同时为0或者1的条件。引入 n 个0-1辅助变量 z_1, z_2, \dots, z_n 。令

$$\sum_{j=1}^m \sum_{t=1}^N X_{ij}^1 + z_i = 1, \quad i=1,2,\dots,n \quad (2)$$

$$\sum_{j=1}^m \sum_{t=1}^N X_{ij}^2 + z_i = 1, \quad i=1,2,\dots,n \quad (3)$$

则当 $z_i=1$ 时, $u_i^1 = u_i^2 = 0$; 当 $z_i=0$ 时, $u_i^1 = u_i^2 = 1$ 。因此根据 u_i 的定义有:

$$u_i = \frac{1}{2}(u_i^1 + u_i^2) \quad (4)$$

2.3.2 约束条件2:传感器的最大工作容量限制

当传感器开始探测目标时,在此后的一段时间 Δt_i^1 或 Δt_i^2 内,传感器必须持续探测目标才有效,在这段时间内传感器资源处于占用状态,必须保证每个时刻传感器 j 同时探测的目标数目不超过最大工作容量 C_j 。用符号 y_{ijt} 表示传感器 j 在时刻 t 对目标 i 的探测情况。 $y_{ijt}=1$ 表示 t 时刻传感器 j 正在探测目标 i 。记 $\mathbf{Y} = (y_{ijt})_{n \times m \times N}$ 表示由 y_{ijt} 构成的三维矩阵。

下面推导 \mathbf{Y} 与 X_{ij}^k 的关系式,考虑 $X_{ij}^k=1$ 表示从 t 时刻到 $t + \Delta t_i^k$ 时刻之前传感器的资源始终被目标 i 占用,即对所有满足 $t \leq s < t + \Delta t_i^k$ 的时刻 s 有 $y_{ijs}=1$ 。用一个 $n \times m \times N$ 的三维矩阵 \mathbf{I}_{ij}^k 表示 $X_{ij}^k=1$ 时的决策对资源的使用情况。当满足 $t \leq s < t + \Delta t_i^k$ 时, $\mathbf{I}_{ij}^k[i, j, s] = 1$, 其余情况取0。每一个 X_{ij}^k 都对应一个 \mathbf{I}_{ij}^k , 共 $2 \times n \times m \times N$ 个。

每一个矩阵 \mathbf{I}_{ij}^k 表示在决策 $X_{ij}^k=1$ 下资源的使用,将所有 $X_{ij}^k=1$ 对应的 \mathbf{I}_{ij}^k 累加,即可得出 \mathbf{Y} :

$$\mathbf{Y} = \sum_{k=1}^2 \sum_{i=1}^n \sum_{j=1}^m \sum_{t=1}^N X_{ij}^k \times \mathbf{I}_{ij}^k \quad (5)$$

由 \mathbf{Y} 的含义知,对于固定的 j 和 t ,使 $y_{ijt}=1$ 的 i 的个数即为传感器 j 在 t 时刻的工作量,其应不大于传感器 j 的最大工作容量 C_j ,即:

$$\sum_{i=1}^n y_{ijt} \leq C_j, \quad j=1,2,\dots,m; t=1,2,\dots,N \quad (6)$$

2.3.3 约束条件3:分段探测的先后时间限制

成功跟踪一个目标需要两段探测并且要保持两段探测的先后关系和时间间隔,即在第一段探测完成之后的 Δt_i 时长之前都不会开始第二段探测。

考虑工作饱和情况下第一段探测和第二段探测的次数均不超过一次(式(2)~(3)),所以对目标 i ,任意时刻 t_1 之后开始第一段探测的次数与 t_1 时刻之前开始第一段探测的次数的差只能取0和 ± 1 。取值为1表示第一段探测是在 t_1 时刻或之后开始,取值为-1表示第一段探测是在 t_1 时刻之前开始,取值为0表示放弃跟踪目标 i 。即:

$$\sum_{j=1}^m \sum_{t \geq t_1}^N X_{ij}^1 - \sum_{j=1}^m \sum_{t < t_1}^N X_{ij}^1 = 0 \text{ 或 } \pm 1 \quad (7)$$

当式(7)取值为1时,表示目标 i 的第一段探测发生在 t_1 时刻之后,为了有足够的时间进行第二段探测也该有 $t_1 + \Delta t_i^1 + \Delta t_i < N - \Delta t_i^2$ 。若满足两段探测的条件,第二段探测至少在 $t_1 + \Delta t_i^1 + \Delta t_i$ 之后开始。在此之前不应该进行第二段探测的任务分配,即:

$$\sum_{j=1}^m \sum_{t < t_1 + \Delta t_i^1 + \Delta t_i}^N X_{ij}^2 = 0 \quad (8)$$

因为第二段探测最多一次,显然有:

$$\sum_{j=1}^m \sum_{t < t_1 + \Delta t_i^1 + \Delta t_i}^N X_{ij}^2 \leq u_i^2 \leq 1 \quad (9)$$

由式(7)和式(9)得:

$$\sum_{j=1}^m \left(\sum_{t \geq t_1}^N X_{ij}^1 - \sum_{t < t_1}^N X_{ij}^1 + \sum_{t < t_1 + \Delta t_i^1 + \Delta t_i}^N X_{ij}^2 \right) \leq 1 \quad (10)$$

式(10)要对所有满足 $t_1 + \Delta t_i^1 + \Delta t_i < N - \Delta t_i^2$ 的 t_1 成立。

下证式(10)也是约束条件3的充分条件。假设式(10)成立,设传感器 j 在 t_1 时刻对目标 i 进行第一段探测,则:

$$\sum_{j=1}^m \sum_{t \geq t_1}^N X_{ij}^1 - \sum_{j=1}^m \sum_{t < t_1}^N X_{ij}^1 = 1 \quad (11)$$

由式(10)得:

$$\sum_{j=1}^m \sum_{t < t_1 + \Delta t_i^1 + \Delta t_i}^N X_{ij}^2 = 0 \quad (12)$$

即 $t_1 + \Delta t_i^1 + \Delta t_i$ 时刻之前没有对目标 i 进行第二段探测,由此可知第二段探测在 $t_1 + \Delta t_i^1 + \Delta t_i$ 时刻之后,充分性得证。综上,式(10)是约束条件3的充要条件。

2.3.4 约束条件4:进入-离开时间限制

若开始探测的时间太靠后,则一段探测未完成目标就可能离开了探测区域,则需要重新选择

开始探测时刻或放弃跟踪。用 T_{ij}^1 表示目标 i 进入传感器 j 区域的时刻,用 T_{ij}^2 表示目标 i 离开传感器 j 区域的时刻。若目标 i 不进入传感器 j 扫描区域,取 $T_{ij}^1 = T_{ij}^2 = 0$ 。

1) 情况 1: $T_{ij}^1 = T_{ij}^2 = 0$, 即目标 i 不进入传感器 j 扫描区域, 令

$$X_{ij}^1 = 0, \quad t = 1, 2, \dots, N \quad (13)$$

$$X_{ij}^2 = 0, \quad t = 1, 2, \dots, N \quad (14)$$

2) 情况 2: ①当 $t + \Delta t_i^1 > T_{ij}^2 + 1$, 即目标 i 第一段探测的结束时间晚于离开传感器 j 扫描区域的时间, 令

$$X_{ij}^1 = 0, \quad t > T_{ij}^2 - \Delta t_i^1 + 1 \quad (15)$$

②类似地, 当 $t + \Delta t_i^2 > T_{ij}^1 + 1$, 令

$$X_{ij}^2 = 0, \quad t > T_{ij}^1 - \Delta t_i^2 + 1 \quad (16)$$

2.4 0-1 线性整数规划模型

为了求解方便, 在整数规划中, 一般习惯用列向量表示决策变量, 用矩阵表示约束条件。将由 X_{ij}^k, z_i 共 $2nmN + n$ 个变量构成的决策变量变换为一维列向量 \mathbf{v} 。将三种变量字典顺序排列: X_{ij}^1 下标对应为 $(i-1)mN + (j-1)N + t$; X_{ij}^2 下标对应为 $(n+i-1)mN + (j-1)N + t$; z_i 下标对应为 $2nmN + i$ 。这样每一个 0-1 变量都在 \mathbf{v} 中有固定的位置。

约束分为等式约束和不等式约束, 分别是式(2)、式(3)、式(13)~(16)六组等式和式(6)、式(10)两组不等式。约束矩阵分为等式约束矩阵 \mathbf{Aeq} 和不等式约束矩阵 \mathbf{A} 。 \mathbf{Aeq} 的各行分别是式(2)、式(3)、式(13)~(16)六组等式中左端决策变量的系数按 \mathbf{v} 中决策变量的顺序排列的行向量。类似地, 约束矩阵 \mathbf{A} 对应于式(6)、式(10)两组不等式。右端向量 \mathbf{beq} 和 \mathbf{b} 分别是 \mathbf{Aeq} 和 \mathbf{A} 对应约束等式和不等式的右侧常数项排列组成的列向量。目标函数为:

$$z = \mathbf{w}^T \mathbf{u} = \frac{1}{2} \sum_{i=1}^n \sum_{k=1}^2 \sum_{j=1}^m \sum_{t=1}^N w_i X_{ij}^k \quad (17)$$

令 \mathbf{c} 表示 \mathbf{v} 中决策变量在目标函数中的系数排列成的列向量, 于是, 模型可以写成标准的线性 0-1 整数规划格式:

$$\begin{aligned} & (\text{IP}) \max \mathbf{c}^T \mathbf{v} \\ & \text{s. t. } \begin{cases} \mathbf{A}\mathbf{v} \leq \mathbf{b}, \\ \mathbf{Aeq} \cdot \mathbf{v} = \mathbf{beq}, \end{cases} \\ & \mathbf{v} \in \{0, 1\}^r, r = 2nmN + n \end{aligned} \quad (18)$$

3 基于模拟退火的求解算法

目前, 用来求解规模较大的优化和调度问题的常用方法是智能搜索算法, 如遗传算法、粒子群

算法、禁忌搜索算法、模拟退火算法等。这些算法能够在可接受的时间内得出近似最优解。同其他的智能搜索算法相比, 模拟退火算法收敛性较好, 理论上可以证明以概率 1 收敛到最优解, 并且具有描述简单、使用灵活、运行效率高和较少受到初始解影响等优点, 因此得到较多关注并应用于一些优化问题, 取得了较好的效果。

3.1 算法的要素

模拟退火算法是一种模拟固体退火过程的算法。使用模拟退火算法时需给出初始温度、终止温度、降温函数、能量函数、解的邻域和移动等算法要素。在本例中降温函数选用 $T_k = T_0 / [\log(1 + k)]$, $k \in \mathbb{N}^+$ 。能量函数是模拟退火算法中表示当前解优劣的函数, 能量越低对应的解越接近于最优。模拟退火算法试图找到使能量函数最小的解。因为原问题中目标是最大化被跟踪目标的重要程度之和, 所以本例中能量函数 f 用未被跟踪的目标的重要程度之和表示。

模型中一个解的邻域是由从这个解对应的调度策略中增减调度方式产生的所有可行调度策略构成的集合。解的移动分为无条件移动和有条件移动。设 i 为当前解, f 为对应能量函数, j 为当前解邻域中的某个解。若 $f(j) < f(i)$, 则当前解可以进行无条件移动。反之, 若 $f(j) > f(i)$, 则当前解 i 以一定概率转移至解 j 。

3.2 算法步骤

类比退火过程, 将能量 E 模拟为目标函数值 f , 温度 T 演化成控制参数 t , 即得到解组合优化问题的模拟退火算法: 由初始解 i 和控制参数初值 T_0 开始, 对当前解重复“产生当前解邻域中的解→计算目标函数差→接受或舍弃”的迭代, 并逐步衰减 T 值, 当 T 低于终止温度时算法终止, 算法终止时的当前历史最优解即为近似最优解, 这是基于蒙特卡洛迭代求解法的一种启发式随机搜索过程。下面是算法的步骤:

步骤 1: 初始化。任选初试解 i , 给定初始温度 T_0 和终止温度 T_f , 确定降温函数 ΔT 和循环次数 $n(T_k)$, 令迭代指标 $k = 0, T_k = T_0$ 。

步骤 2: 随机产生 i 的一个邻域解 j , 计算目标值增量 $\Delta f = f(j) - f(i)$ 。

步骤 3: 若 $\Delta f < 0$, 令 $i := j$, 转步骤 4; 否则, 产生随机数 $\xi \in (0, 1)$, 若 $\exp\left(\frac{\Delta f}{T_k}\right) > \xi$, 则令 $i := j$ 。

步骤 4: 若达到热平衡 (内循环次数大于 $n(T_k)$), 转步骤 5; 否则转步骤 2。

步骤5:按降温函数降低温度 $T_k, k: = k + 1$, 若 $T_k < T_f$, 停止迭代; 否则转步骤2。

4 模拟实验及结果分析

4.1 实例构建

为进行仿真实验, 构建一个由3个传感器

对24个目标进行跟踪的资源分配问题实例。其中, 表1描述了模型的基本参数; 表2根据目标所在的探测区域不同将目标分成六组, 并列出了每组目标的重要程度; 表3给出了上述每一组目标在不同传感器探测区域的进入和离开时刻。

表1 基本参数

Tab.1 Basic parameters

目标数目	传感器数目	时间段数	最大工作容量	第一段探测最短时间	第二段探测最短时间	时间间隔
$n = 24$	$m = 3$	$N = 11$	$C_j = 2$	$\Delta t_i^1 = 2 \text{ s}$	$\Delta t_i^2 = 3 \text{ s}$	$\Delta t_i = 0 \text{ s}$

表2 目标重要程度

Tab.2 Weight of targets

目标编号	1	2	3	4	5	6	7	8	9	10	11	12
重要程度	1	2	3	4	1	2	3	4	1	2	3	4
目标编号	13	14	15	16	17	18	19	20	21	22	23	24
重要程度	1	2	3	4	1	2	3	4	1	2	3	4

表3 目标进入和离开传感器探测区域的时刻表(t_{ij}^1, t_{ij}^2)

Tab.3 Entry time and leave time of targets (t_{ij}^1, t_{ij}^2)

	目标1	目标2	目标3	目标4	目标5	目标6	目标7	目标8
传感器1	(1,4)	(1,4)	(1,4)	(1,4)	(1,11)	(1,11)	(1,11)	(1,11)
传感器2	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)
传感器3	(1,11)	(1,11)	(1,11)	(1,11)	(0,0)	(0,0)	(0,0)	(0,0)
	目标9	目标10	目标11	目标12	目标13	目标14	目标15	目标16
传感器1	(1,5)	(1,5)	(1,5)	(1,5)	(0,0)	(0,0)	(0,0)	(0,0)
传感器2	(1,11)	(1,11)	(1,11)	(1,11)	(1,11)	(1,11)	(1,11)	(1,11)
传感器3	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)
	目标17	目标18	目标19	目标20	目标21	目标22	目标23	目标24
传感器1	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)
传感器2	(1,2)	(1,2)	(1,2)	(1,2)	(0,0)	(0,0)	(0,0)	(0,0)
传感器3	(1,11)	(1,11)	(1,11)	(1,11)	(1,11)	(1,11)	(1,11)	(1,11)

4.2 割平面法与模拟退火算法实验结果对比分析

利用上述实例数据进行了仿真模拟, 用割平面法求解式(18)所示0-1线性整数规划模型。调用MATLAB软件包求解, 得到传感器对目标的跟踪策略, 如表4所示。

从表4看出, 按最优策略一共成功跟踪了13个目标, 跟踪目标重要程度之和为44。在式(18)所示模型中, 去掉0-1整数约束, 得到一个松弛的线性规划模型, 利用单纯形法求解

得到目标函数最优值上界为44.4。注意前面提出的0-1整数规划模型其目标函数值是整数, 这说明, 表4的传感器调度策略是最优策略。结合表3也可以看出重要程度大的目标均被跟踪。表5记录了最优调度策略下各传感器对各目标的具体跟踪情况。

表5中相邻单元格表示传感器对一个目标在多个时间段内连续的探测。可以看出, 只有传感器1在时刻8的资源没有被使用, 其余时间所有

传感器处于最大工作状态;由于每个目标需要 5 个时间段进行跟踪,资源的调度已经非常充分,并且满足了跟踪需要有前后两次探测的要求;没有出现无效的跟踪,如只探测一段或探测时间少于最短时间的情况。

通过上面的分析验证了模型的正确性,说明

该模型准确地描述了传感器资源调度实际问题。下面再用模拟退火算法对该模型求解进行比较分析,将目标的跟踪情况记录于表 6。

从表 6 可以看出,最优策略下成功跟踪了 12 个目标,跟踪目标重要程度之和为 41。

表 7 记录了传感器的资源调度情况。

表 4 跟踪情况——割平面法

Tab.4 Tracking status—cutting plane algorithm

目标	1	2	3	4	5	6	7	8	9	10	11	12
跟踪	否	否	是	是	否	是	是	是	否	否	是	是
目标	13	14	15	16	17	18	19	20	21	22	23	24
跟踪	否	否	是	是	否	否	是	是	否	否	是	是

表 5 传感器任务分配——割平面法

Tab.5 Task allocation of sensors—cutting plane algorithm

传感器	各时间段探测的目标										
	1	2	3	4	5	6	7	8	9	10	11
传感器 3	目标 23	目标 23	目标 19	目标 19	目标 19	目标 20	目标 20	目标 20	目标 3	目标 3	目标 3
	目标 24	目标 24	目标 23	目标 23	目标 23	目标 24	目标 24	目标 24	目标 4	目标 4	目标 4
传感器 2	目标 19	目标 19	目标 11	目标 11	目标 11	目标 12	目标 12	目标 12	目标 15	目标 15	目标 15
	目标 20	目标 20	目标 12	目标 12	目标 15	目标 15	目标 16	目标 16	目标 16	目标 16	目标 16
传感器 1	目标 6	目标 6	目标 3	目标 3	目标 6	目标 6	目标 6	未使用	目标 7	目标 7	目标 7
	目标 11	目标 11	目标 4	目标 4	目标 8	目标 8	目标 7	目标 7	目标 8	目标 8	目标 8

表 6 跟踪情况——模拟退火算法

Tab.6 Tracking status—simulated annealing algorithm

目标	1	2	3	4	5	6	7	8	9	10	11	12
跟踪	否	否	是	是	否	否	是	是	否	是	是	是
目标	13	14	15	16	17	18	19	20	21	22	23	24
跟踪	否	否	是	是	否	否	否	是	否	否	是	是

表 7 传感器任务分配——模拟退火算法

Tab.7 Tack allocation of sensors—simulated annealing algorithm

传感器	各时间段探测的目标										
	1	2	3	4	5	6	7	8	9	10	11
传感器 3	目标 20	目标 20	目标 4	目标 4	目标 4	目标 3	目标 3	目标 3	目标 20	目标 20	目标 20
	目标 24	目标 24	未使用	目标 19	目标 19	目标 19	目标 19	目标 19	目标 24	目标 24	目标 24
传感器 2	目标 10	目标 10	目标 10	目标 10	目标 10	未使用	目标 15	目标 15	目标 15	目标 15	目标 15
	目标 11	目标 11	目标 11	目标 11	目标 11	未使用	目标 16	目标 16	目标 16	目标 16	目标 16
传感器 1	目标 4	目标 4	目标 3	目标 3	未使用	未使用	目标 7	目标 7	目标 7	目标 7	目标 7
	目标 11	目标 11	目标 11	目标 11	目标 11	未使用	目标 8	目标 8	目标 8	目标 8	目标 8

图1是两种调度结果的甘特图。设目标编号是 M_i ,甘特图中 M_i01 表示第一段探测, M_i02 表示第二段探测。从图1可以看出,割平面法的最优策略下只有1个资源被浪费,而由模拟退火算法得到的策略中增加了5个被浪费的资源。因此,由模拟退火算法得到的调度策略的资源使用效率相比于最优调度策略仍然较低,并且目标函数最优值上也有一定的差距。但是在大规模实际问题的求解时间上,启发式算法的优势得到了体现。事实上,大规模情况中割平面法往往在规定的时间内得不到可行解,但启发式算法求解的时间长度并没有随问题规模增大而显著增加,并且其还能够较快地得到一个较优的解。在实际问题中,跟踪的目标数量应在100个左右,从调度开始到结束时间在800s左右。由于运算精度的限制,时间段划分则至少在400段以上,要求算法在

20s内计算出最佳调度策略,这对算法的求解速度有很高的要求。从算法运行时间的对比可以看出,虽然模拟退火算法求解的目标函数值与最优值有一定的差距,但是在求解时间上有巨大的优势,已经基本满足了实际问题的需要。

一次调度在某一时间段内的资源利用率是指该时间段内使用的传感器资源的数目与总传感器的资源数目之比(每个传感器 j 有 C_j 个资源)。资源利用率指标可以直观地反映出一次调度的效率,资源利用率越高,调度的效率越高。图2表示的是两种算法求得的调度结果在每一时间段内的资源利用率,即资源利用率曲线。



图1 甘特图
Fig.1 Gantt chart

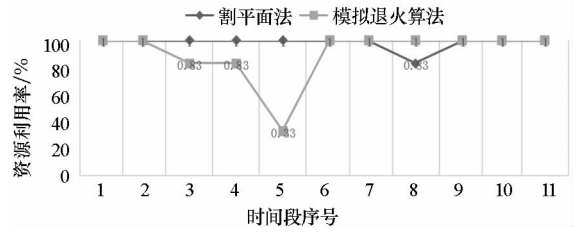


图2 资源利用率对比图

Fig.2 Comparison diagram of resource utilization

从图2可以看出割平面法产生的调度在资源使用上的优势,几乎每个时间段内全部的传感器资源都能被有效地调度起来,而对于模拟退火算法产生的调度,在时间段2至时间段6中出现了许多的资源浪费。

对于时间精度要求更高的模型,即时间段数划分更大时,启发式算法显示出求解速度快的优势。下面对表8中列出的基本参数的模型用割平面法和模拟退火算法求解,以对比两种算法的计算速度。

表8 基本参数

Tab.8 Basic parameters

目标数目	传感器数目	时间段数	最大工作容量	第一段探测最短时间	第二段探测最短时间	时间间隔
$n = 80$	$m = 5$	$N = 400$	$C_j = 8$	$\Delta t_i^1 = 80 \text{ s}$	$\Delta t_i^2 = 100 \text{ s}$	$\Delta t_i = 20 \text{ s}$

在同一台PC上对表8中的模型求解,模拟退火算法速度是割平面法的10倍以上,并随着数据规模的增大,倍数会扩大。图3是使用模拟退火算法得到的资源利用率曲线。其中,最晚截止时间被控制在 $t = 340$,即在时间段340以后传感器不再跟踪目标。

为了仿真实验构建了一组不同规模的模拟实例,每个模拟实例都分别用模拟退火算法和割平面算法求解,运行时间、求解得到的最优值以及跟踪目标数于表9,其中SA表示模拟退火算法,CP

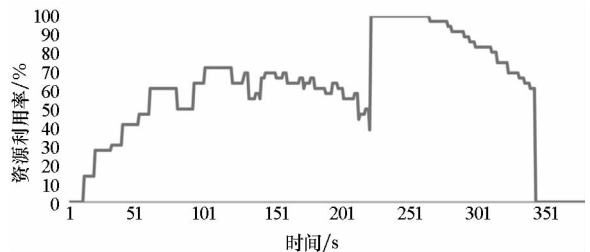


图3 模拟退火算法资源利用率图

Fig.3 Resource utilization of simulated annealing algorithm

表示割平面法。运行计算机的配置是 2.7 GHz Intel Core i5 处理器,8 G 内存。表格中案例 5 至案例 8 在利用割平面方法求解时内存超出限制,只能用启发式算法求解。

从表 9 可以看出模拟退火算法的运行时间随规模的增大并没有显著增加,也满足了规模为 100 个目标的实际问题中运行时间不超过 20 s 的限制。

表 9 运行时间和结果对比
Tab.9 Comparison of running time and results

规模		40 个目标		80 个目标		160 个目标		240 个目标	
案例标号		1	2	3	4	5	6	7	8
运行时间/s	SA	4	4	8	7	17	17	34	33
	CP	31	31	360	362	—	—	—	—
目标函数值	SA	32	44	78	64	70	86	138	119
	CP	34	47	72	74	—	—	—	—
跟踪目标数	SA	12	18	29	24	24	30	48	37
	CP	14	21	32	27	—	—	—	—

4.3 模拟退火算法与遗传算法实验结果对比分析

柔性作业车间调度问题常用遗传算法求解^[13-14],受此启发,将两段探测目标的传感器调度问题模型转化为一个特殊的柔性作业车间调度问题,用遗传算法求解,并与模拟退火算法对比。

传感器对一个目标的两段探测对应两道工序。将每个传感器 j 看作是 $C_j(j = 1, 2, \dots, m)$ 个子机器的并集,每个子机器只能加工一个工件。目标被传感器跟踪时前后两段探测的探测时间对应于工件被机器加工时两道工序所需的加工时间。工件与其对应的目标有相同的重要程度。取遗传算法的适应值为在截止时间之前加工完成的工件的重要程度之和。

用模拟退火算法和遗传算法对四个模拟案例进行求解,结果见表 10。由表 10 可看出,遗传算法无论是在求解时间还是目标函数值方面效果均不如模拟退火算法。事实上,在遗传算法中,目标函数为在截止时间前被加工完成的目标的重要程

度之和,但是解码时整串编码都需要先转为一个调度,包括在截止时间后完成加工的目标,因此降低了算法的求解效率。

5 结论

双段探测目标的传感器资源调度问题是一种特殊的资源调度问题。本文以跟踪目标的探测次数、时间和传感器资源等为约束条件建立了 0-1 线性整数规划模型,从整数规划的角度对模型进行分析和求解,同时提出了对应的模拟退火算法。通过模拟退火算法与割平面法及遗传算法的对比发现模拟退火算法求解这类问题时在时间和效率上具有明显优势。对于小规模问题,割平面法能得到全局最优解,便于理论分析。提出的模型与调度算法在实际中能为双段探测目标的传感器资源调度问题提供一个可行的方案。如何借鉴车间调度问题的研究方法来解决双段探测目标的传感器资源调度问题值得进一步研究。

参考文献 (References)

[1] 张伯彦,蔡庆宇. 相控雷达阵的自适应调度和多目标数据处理技术[J]. 电子学报, 1997, 25(9): 1-5.
ZHANG Boyan, CAI Qingyu. Adaptive scheduling and multitarget data processing techniques of phased array radars[J]. Acta Electronica Sinica, 1997, 25(9): 1-5. (in Chinese)

[2] 李昊,于周秋. 基于优先级的相控阵测量雷达调度设计[J]. 现代雷达, 2006, 28(7): 52-55.
LI Hao, YU Zhouqiu. Design of phased array instrumentation radar scheduling algorithm based on priority[J]. Modern Radar, 2006, 28(7): 52-55. (in Chinese)

[3] 孟宪福. 基于优先级的任务调度与负载均衡模型研究[J]. 小型微型计算机系统, 2005, 26(9): 1601-1605.
MENG Xianfu. Study of task scheduling and load balancing

表 10 两种启发式算法的对比

Tab.10 Comparison of two heuristic algorithms

案例	案例	9	10	11	12
规模	目标数	240	160	160	80
模拟退火方法	跟踪目标数	44	29	24	12
	目标函数值	127	83	72	36
	运行时间/s	37	17	17	9
遗传算法	跟踪目标数	38	26	23	12
	目标函数值	86	62	54	30
	运行时间/s	116	68	66	34

- models based on priority [J]. *Mini-Micro Systems*, 2005, 26(9): 1601 - 1605. (in Chinese)
- [4] Huizing A G, Bloemen A A F. An efficient scheduling algorithm for a multifunction radar [C]// *Proceedings of International Symposium on Phased Array Systems and Technology*, 1996: 359 - 364.
- [5] 何金新, 邱杰, 王国宏. 相控阵雷达事件调度中的时间窗研究[J]. *雷达科学与技术*, 2010, 8(1): 80 - 86.
HE Jinxin, QIU Jie, WANG Guohong. Study on time window in multifunction phased array radar task scheduling [J]. *Radar Science and Technology*, 2010, 8(1): 80 - 86. (in Chinese)
- [6] Farina A, Neri P. Multi-target interleaved tracking for phased-array[J]. *IEE Proceedings, Part F—Communications, Radar and Signal Processing*, 1980, 127(4): 312 - 318.
- [7] Shih C S, Gopalakrishnan S, Ganti P. Scheduling real-time dwells using tasks with synthetic periods[C]//*Proceedings of 24th IEEE International Real-Time Systems Symposium*, 2003: 210 - 219.
- [8] Shih C S, Ganti P, Gopalakrishnan S. Synthesizing task periods for dwells in multi-function phased array radars[C]//*Proceedings of the IEEE Radar Conference, Philadel*, 2004: 145 - 150.
- [9] Gopalakrishnan S, Caccamo M, Shih C S. Finite-horizon scheduling of radar dwells with online template construction[C]//*Proceedings of IEEE International Real-Time Systems Symposium*, 2004: 23 - 33.
- [10] 陈怡君, 罗迎, 张群, 等. 基于认知 ISAR 成像的相控阵雷达资源自适应调度算法[J]. *电子与信息学报*, 2014, 36(7): 1566 - 1572.
- CHEN Yijun, LUO Ying, ZHANG Qun, et al. Adaptive scheduling algorithm for phased array radar based on cognitive ISAR imaging [J]. *Journal of Electronics & Information Technology*, 2014, 36(7): 1566 - 1572. (in Chinese)
- [11] 程小枫, 涂刚毅, 吴少鹏. 双波段旋转相控阵雷达任务调度算法[J]. *科学技术与工程*, 2014, 14(23): 73 - 80.
CHENG Xiaofeng, TU Gangyi, WU Shaopeng. Task scheduling algorithm for dual-band rotating phased array radar[J]. *Science Technology and Engineering*, 2014, 14(23): 73 - 80. (in Chinese)
- [12] 高亮, 张国辉, 王晓娟. 柔性作业车间调度智能算法及其应用[M]. 1版. 武汉: 华中科技大学出版社, 2012.
GAO Liang, ZHANG Guohui, WANG Xiaojuan. *Flexible shop scheduling intelligent algorithm and its application*[M]. 1st ed. Wuhan: Huazhong University of Science and Technology, 2012. (in Chinese)
- [13] 王万良, 吴启迪, 宋毅. 求解作业车间调度问题的改进自适应遗传算法[J]. *系统工程理论与实践*, 2004, 24(2): 58 - 62.
WANG Wanliang, WU Qidi, SONG Yi. Modified adaptive genetic algorithms for solving job-shop scheduling problems[J]. *Systems Engineering—Theory & Practice*, 2004, 24(2): 58 - 62. (in Chinese)
- [14] 张超勇, 饶运清, 李培根, 等. 柔性作业车间调度问题的两级遗传算法[J]. *机械工程学报*, 2007, 43(4): 119 - 124.
ZHANG Chaoyong, RAO Yunqing, LI Peigen, et al. Bilevel genetic algorithm for the flexible job-shop scheduling problem[J]. *Chinese Journal of Mechanical Engineering*, 2007, 43(4): 119 - 124. (in Chinese)