

基于 SSA 算法的 TTE 静态调度表求解方法

刘冉, 卢满宏, 张显辉, 肖佳洛, 杨文涛
(北京遥测技术研究所 北京 100076)

摘要: 当前网络实时系统飞速发展, 为了能达到更高效的通信, 网络资源的调度问题受到专家学者的广泛关注。本文所研究的静态调度表求解方法是配置网络资源的有效解决方案, 是国内外相关领域的研究重点。自 2002 年 TTE (Time-Triggered Ethernet, 时间触发以太网) 问世以来, 由于应用领域和具体使用场景的不同, TTE 网络静态调度表求解方法也在不断改进创新。随着各类算法的不断优化, 静态调度表的生成效果越来越理想, 但方案仍然不能很好地应用于工程实践。本文首次提出选用 SSA 算法 (Sparrow Search Algorithm, 麻雀搜索算法), 并在个体最优位置中加入扰动, 使得适应度函数的收敛效率有了明显改善, 避免陷入局部最优, 达到了求解 TTE 静态调度表的目的。

关键词: 麻雀搜索算法; 时间触发以太网; 静态调度表

中图分类号: TP393.1 **文献标志码:** A **文章编号:** 2095-1000(2025)01-0107-08

DOI: 10.12347/j.ycyk.20240806001

引用格式: 刘冉, 卢满宏, 张显辉, 等. 基于 SSA 算法的 TTE 静态调度表求解方法[J]. 遥测遥控, 2025, 46(1): 107-114.

Static Scheduling Pattern of Time-Triggered Ethernet Based on SSA Algorithm

LIU Ran, LU Manhong, ZHANG Xianhui, XIAO Jialuo, YANG Wentao
(Beijing Research Institute of Telemetry, Beijing 100076, China)

Abstract: With the rapid development of network real-time system, in order to achieve more efficient communication, the problem of network resource scheduling has been widely concerned by experts and scholars. The static scheduling table is an effective solution for configuring network resources, which is the focus of research in related fields at home and abroad. Since the advent of TTE (Time-Triggered Ethernet) in 2002, the solution method of TTE network static scheduling table has been improved and innovated because of different application fields and specific use scenarios. With the continuous optimization of various algorithms, the generating effect of static scheduling table is more and more ideal, but the scheme still can not be applied to engineering practice. This paper proposes to select Sparrow Search Algorithm (SSA) and add disturbance to the individual optimal position, which can significantly improve the convergence efficiency of fitness function, avoid falling into local optimal, and achieve the purpose of solving TTE static scheduling table.

Keywords: Sparrow search algorithm; Time triggered ethernet; Static scheduling table

Citation: LIU Ran, LU Manhong, ZHANG Xianhui, et al. Static Scheduling Pattern of Time-Triggered Ethernet Based on SSA Algorithm[J]. Journal of Telemetry, Tracking and Command, 2025, 46(1): 107-114.

0 引言

当前, 随着网络规模的不断扩大, 通信设备越来越多, 如何合理地安排调度设备间收发消息成为国内外相关领域的研究重点^[1]。本文所研究的 TTE(Time Triggered Ethernet, 时间触发以太网)总线, 由于其本身特性, 节点间的数据消息采用时间触发机制进行通信, 因此需要管理者根据消息传输周期和携带数据长度, 来建立全网统一的时

间调度表。

目前, 为保证航空航天领域设备间传输的数据消息的实时性, 多为预先规划设置好的确定性消息, 因此本文的主要研究方向为如何在程序运行之前合理生成各终端的调度时间表, 即求解静态调度表。此问题的解决方案主要有: 以求解优化方程为方向的调度方法和以求解启发式算法为方向的调度方法。

以求解优化方程为方向的调度方法主要是通

过求解器并设置约束条件来快速求得最优解^[2]。从求解原理来看, 求解器通过轮询问答的方式进行求解, 计算资源消耗极大^[3]; 从工程实际应用来看, 一旦网络设备增加或网络拓扑结构复杂时, 该算法可能会难以完成求解, 且无法解决大时延或消息抖动等问题。

以求解启发式算法为方向的调度方法是预先规划好不同消息的发送周期, 计算出数据链路上的总负载, 通过 RMS(Rate Monotonic Scheduling, 单调速率调度算法)机制对数据消息进行优先级排序, 合理分配网络资源^[4]。由于该算法需要预设消息的传输周期, 其求解难度还与消息长度有很大的相关性, 因此在该方法下难以增加网络设备且无法适用于多跳网络。

2019 年, 商行^[5]提出了一个融合负载均衡理念的带宽分配框架, 同时构建了应用层与链路层间的协同约束机制。基于这一框架, 通过深度融合启发式策略与优化数学模型的思想, 创造出一种旨在最大化降低系统响应延迟的负载均衡导向型调度表生成策略。

2021 年, 张春韵^[6]等针对电力经济调度问题提出了使用改进的粒子群优化算法, 电力系统经济调度也是一种多约束、非线性问题。文章不仅将改进粒子群算法应用于 8 种案例中, 证明了其具有优越的收敛性和稳定性, 还提出一种凹函数模型的动态惯性权重法, 提高了算法的全局搜索能力。

2022 年, 范雅男等将基于位置相似度的禁忌搜索与粒子群优化算法结合形成一种新的算法, 应用于流水车间调度问题中, 并引入自适应惯性权重因子和扰动因子更新速度^[5], 避免了算法陷入局部最优解。同时在种群初始化时采用 NEH(用于解决流生产线调度问题的启发式算法)算法, 使得粒子更具多样性。

2023 年, 陈春燕^[6]等人紧随人工智能发展浪潮, 提出一种基于 Q 学习理论的调度规划策略。他们将原本一维时间线上的消息调度问题, 扩展至三维空间的多宝箱探索模型中, 进而利用强化学习, 实现了静态调度表的有效求解。

2024 年, 南京邮电大学的刘齐坚提出一种使用黄金正弦与莱维飞行机制的麻雀搜索算法。其中所具有的创新性弥补了之前该算法在异构多核处理器任务调度领域应用中的空白^[7]。该算法在满足任务间通信约束条件下, 最优化任务间的调度

长度进而迭代寻优, 能够有效缩短任务调度的时间。

纵观国内外针对静态调度表生成策略的研究现状发现, 尽管这些研究针对特定应用环境展现出了良好性能, 但并不具有普适性。鉴于此, 本文致力于研究出更为普适的静态调度表生成方案, 将通信过程中的各种资源约束用数学模型表示, 通过自适应权重因子来表示实际的工程环境。同时创造性地提出了基于 SSA 算法的 TTE 静态调度表求解, 并在个体最优位置加入扰动, 具有更好的收敛效果。

1 TTE 静态调度表

1.1 问题提出

求解静态调度表问题实际上是求解多项式复杂程度的非确定性问题, 其求解复杂程度与通信网络中节点数量、网络拓扑结构、链路上的约束条件等密切相关^[8]。查阅相关研究文献发现, 仿真生成静态调度表随着数据帧数量增加、数据长度改变而求解难度变大, 资源消耗急剧增长, 工程应用中更是应用需求纷乱而难以对众多约束进行数学建模。因此, 如何科学表示通信链路中的资源冲突, 并选择高效且易于实现的调度表生成方法, 是当前的研究重点。

1.2 构造模型

TTE 网络中有三种数据帧传输, 分别为 TT(Time-triggered Traffic, 时间触发)帧、RC(Rate-Tonstrained, 速率受限)帧和 BE(Best-effort, 尽力而为)帧^[9], 其中只有 TT 帧周期性时间触发, 因此本文只研究 TT 帧的静态调度表生成方案。将通信网络中的节点和数据链路上约束条件转化成数学模型, 方便求解静态调度表。

用 $G(N, D)$ 表示 TTE 通信网络, 其中 N 是网络中所有设备的集合, 包括端系统和交换机; D 是网络设备间通信链路的集合。

数据链路定义为相邻节点间相反数据传输方向的两条链路, 用向量 L 表示, 公式表达如下:

$$\forall n_1, n_2 \in N; (n_1, n_2) \in D \Rightarrow [n_1, n_2] \in L, [n_2, n_1] \in L \quad (1)$$

向量 P 定义为数据流路径的集合, 则从发送端 n_1 到接收端 n_r 的数据流路径 p_i , 公式表达如下:

$$p_i = \{[n_1, n_2], \dots, [n_{r-1}, n_r]\} \quad (2)$$

向量 T 定义为通信链路中 TT 帧的集合, t_i 表示该集中的某帧数据, 则 $t_i^{[n_j, n_k]}$ 为数据链路 $[n_j, n_k]$ 上的 TT 帧 t_i 。

定义虚拟链路 VL 为某发送终端和其所有接收终端之间的数据流路径的集合, 则发送终端为 $first(t_i)$, 接收终端为 $last(t_i)$ 。

在求解静态调度表问题中, 主要关注 TT 帧的三个特性: 发送周期 $t_i.period$, 数据长度 $t_i.length$ 和每条链路上的偏移量 $t_i^{[n_j, n_k]}.offset$ 。其中发送周期和长度都是 TT 帧本身特性, 与实际应用场景有关, 默认为固定值且事先已知。而偏移量 $t_i^{[n_j, n_k]}.offset$ 表示 TT 帧在不同数据链路上有不同的发送时间点, 可用如下公式表示:

$$t_i^{[n_j, n_k]} = \{t_i.period, t_i^{[n_j, n_k]}.offset, t_i.length\} \quad (3)$$

因此, 求解静态调度表问题转化为求解出所有 TT 帧在所有数据链路 L 上的偏移量 $t_i^{[n_j, n_k]}.offset$ 。

根据 AS6802 协议中规定, 集群周期(Cluster Cycle, CC)表示所有 TT 帧周期的最小公倍数, 整合周期(Integration Cycle, IC)表示所有 TT 帧周期

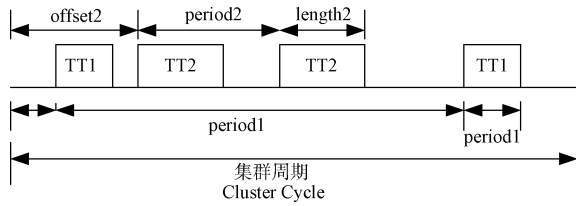


图1 TT帧的各特征值示意图

Fig 1 Diagram of each characteristic value of TT frame

$$\forall [n_j, n_k] \in L, \forall t_m, t_n \in T, \forall a \in [0 \cdots A], \forall b \in [0 \cdots B],$$

$$A = \frac{CC(T.period)}{t_m.period} - 1, B = \left(\frac{CC(T.period)}{t_n.period} - 1 \right),$$

$$\left((t_m \neq t_n) \wedge \exists t_m^{[n_j, n_k]} \wedge \exists t_n^{[n_j, n_k]} \right) \Rightarrow$$

$$\delta_{1m,n}^{[n_j, n_k]}(a, b) = \left| \left(a \times t_m.period + t_m^{[n_j, n_k]}.offset \right) - \left(b \times t_n.period + t_n^{[n_j, n_k]}.offset \right) \right|, \quad (4)$$

$$\phi_{1m,n}^{[n_j, n_k]}(a, b) = \begin{cases} 0, & \delta_{1m,n}^{[n_j, n_k]}(a, b) \neq 0 \\ \theta, & \delta_{1m,n}^{[n_j, n_k]}(a, b) = 0 \end{cases}$$

$$pf_{1m,n}^{[n_j, n_k]} = \sum_{a=0}^A \sum_{b=0}^B \phi_{1m,n}^{[n_j, n_k]}(a, b)$$

其中, $CC(T.period)$ 表示集群周期, $\delta_{1m,n}^{[n_j, n_k]}(a, b)$ 衡量在 $[n_j, n_k]$ 链路上帧 t_m 的第 a 周期和帧 t_n 的第 b 周期的发送时间点关系。当 $\delta_{1m,n}^{[n_j, n_k]}(a, b)$ 不为 0 时, 说

明两帧没有同时处于同一“时间槽”中, 代价值 $\phi_{1m,n}^{[n_j, n_k]}(a, b)$ 为 0; 当 $\delta_{1m,n}^{[n_j, n_k]}(a, b)$ 等于 0 时, 说明有资源链路冲突。对代价值 $\phi_{1m,n}^{[n_j, n_k]}(a, b)$ 中的 a 和 b 求和

1.3 TTE 调度表约束

1.3.1 时间槽概念

为了优化静态调度表的构建过程, 引入等长时隙的概念^[11], 如图 2 所示, 即将一个集群周期划分为多个等长的时段, 称为“时间槽”。在这一设定下, 每个时间槽中能够承载且仅承载一帧 TT 数据, 从而确保资源分配的高效有序。在引入时间槽后, 数据帧长度 $t_i.length$ 可以不予考虑。

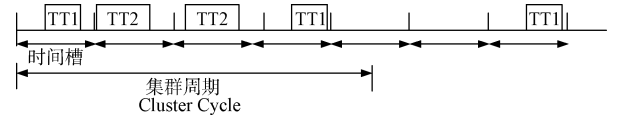


图2 时间槽概念图

Fig 2 The concept map of time slot

1.3.2 适应度函数设计

为了防止 TT 帧在通信链路中发生资源冲突, 本小节推导了影响数据传输的相关因素, 并将这些冲突通过数学公式表示, 方便之后的建模优化。

① 无竞争约束(Contention-Free Constraints)

该约束规定了两个 TT 帧不可同时处于同一个“时间槽”中, 即系统的每一个节点需要逐次处理每一帧数据, 该约束用式(4)表示:

得到所需要的代价函数 $pf_{1m}^{[n_i, n_k]}$, 进而对 TT 帧和数据链路求和, 得到无竞争约束下的损失函数 pf_1 , 如式(5)所示。

$$pf_1 = \sum_{[n_i, n_k]}^L \sum_{t_m, t_n}^T pf_{1m}^{[n_i, n_k]} \quad (5)$$

② 中继节点约束(Trunk-Node Constraints)

中继节点约束包括路径传输约束和交换机内存约束, 用式(6)表示。

路径传输约束规定了每一条 TT 帧通过交换机转发时, 当接收消息的数据链路发送完成信号后, 发送消息的数据链路才会继续发送下一帧消息。

交换机内存约束规定了交换机的存储器容量, 即交换机本身的最大延迟时间。

$$\forall vl \in VL, \forall p_i \in vl, \forall [n_x, n_j], [n_j, n_y] \in p_i: \quad (6)$$

$$\delta_{2m}^{[[n_x, n_j], [n_x, n_j]]} = t_m^{[n_x, n_j]}.offset - t_m^{[n_x, n_j]}.offset$$

$$pf_{2m}^{[[n_x, n_j], [n_x, n_j]]} = \begin{cases} 0, H \leq \delta_{2m}^{[[n_x, n_j], [n_x, n_j]]} \leq M; \\ H - \delta_{2m}^{[[n_x, n_j], [n_x, n_j]]}, \delta_{2m}^{[[n_x, n_j], [n_x, n_j]]} < H; \\ \delta_{2m}^{[[n_x, n_j], [n_x, n_j]]} - M, \delta_{2m}^{[[n_x, n_j], [n_x, n_j]]} > M; \end{cases}$$

其中, $[n_x, n_j]$ 和 $[n_j, n_y]$ 为与交换机 n_j 相邻的链路, 而 $\delta_{2m}^{[[n_x, n_j], [n_x, n_j]]}$ 表示某 TT 帧 t_m 在这两条链路上发送时间点的差值, H 和 M 是根据实际路径传输约束和交换机内存约束所设置的上下限。越满足约束条件, 代价函数值越小, 最理想结果是代价函数值 $pf_{2m}^{[[n_x, n_j], [n_x, n_j]]}$ 为 0, 进而对变量求和, 得到中继节点约束如式(7)所示。

$$pf_2 = \sum_{t_m}^T \sum_{[n_x, n_j], [n_x, n_j]}^{p_i} pf_{2m}^{[[n_x, n_j], [n_x, n_j]]} \quad (7)$$

③ 端到端传输约束(End-to-End Transmission Constraints)

依据不同的应用场景定义 TT 帧 t_m 允许的最大端到端延迟。假设一台设备给多台设备传输消息时, 接收设备的接收时延都是相同的, 则此约束可用式(8)表示:

$$\forall t_m \in T, \forall [n_k, n_j] \in last(t_m):$$

$$\delta_{3m}^{[n_k, n_j]} = t_m^{[n_k, n_j]}.offset - t_m^{first(t_m)};$$

$$pf_{3m}^{[n_k, n_j]} = \begin{cases} 0, \delta_{3m}^{[n_k, n_j]} < latency; \\ \delta_{3m}^{[n_k, n_j]} - lat, \delta_{3m}^{[n_k, n_j]} \geq latency; \end{cases} \quad (8)$$

其中, $\delta_{3m}^{[n_k, n_j]}$ 表示某 TT 帧在一个集群周期内全部的传输路径上, 其在最后一条数据链路上的发送时间点与最初的发送时间点之间的差值, 比较该差

值与最大端到端延迟 $latency$ 的大小关系, 定义了代价函数 $pf_{3m}^{[n_k, n_j]}$ 。越满足约束条件, 代价函数值越小, 最理想结果是代价函数值为 0。进而对变量求和, 得到端到端传输约束如式(9)所示。

$$pf_3 = \sum_{t_i}^T \sum_{[n_k, n_j]}^{last(t_i)} pf_{3m}^{[n_k, n_j]} \quad (9)$$

④ 适应度函数设计

将式(5)、式(7)和式(9)结合, 得到 SSA 算法所需要的适应度函数, 如式(10)所示。

$$pf = \omega_1 \times pf_1 + \omega_2 \times pf_2 + \omega_3 \times pf_3 \quad (10)$$

其中 ω_1 、 ω_2 和 ω_3 为常数, 在实际寻优过程中, 随着迭代次数的增加, 各约束函数会呈现出较大的差异。通过调整超参的大小, 可以干涉不同约束的优化力度。本文不断计算每种约束在总约束里所占比重, 然后将占比大的约束分配较小的权重, 占比小的约束分配较大的权重, 最终得到最优适应度函数。

2 基于 SSA 算法的 TTE 静态调度表求解方法

2.1 SSA 算法基本思想

该算法根据麻雀现实中的行为而提出, 扮演不同角色的麻雀会进行位置转换, 从而达到对所求参数的自适应寻优。同时加入警戒机制, 当麻雀个体意识到危险时, 在群体边缘的麻雀会迅速移动到更好位置来躲避捕食者^[12], 因此该算法可以不断进行全局搜索, 避免在算法初期就在局部得到最优解。对比粒子群算法和灰狼优化算法等优化算法, SSA 算法精度更高, 收敛性更好, 收敛速度更快, 具有出色的稳定性, 且易于实现, 对于 TTE 网络的静态调度表求解有很好的适用性。

2.2 SSA 算法流程

在麻雀生活集群中, 可以根据麻雀行为进行角色划分: ① 能量储备高的麻雀有能力去寻找食物, 称为探索者, 负责找到可以提供食物来源的区域; ② 能量储备低的麻雀为追随者, 随着发现者的方向寻找食物; 个体适应度的值即代表了麻雀个体饥饿度的大小。同时, 当麻雀发现附近存在捕食者而感到危险时, 便会作为警戒者, 发出危险信号并放弃觅食^[13]。

麻雀的位置可由如下矩阵表示:

$$X = \begin{bmatrix} X_{1,1} & X_{1,2} & \cdots & \cdots & X_{1,d} \\ X_{2,1} & X_{2,2} & \cdots & \cdots & X_{2,d} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ X_{n,1} & X_{n,2} & \cdots & \cdots & X_{n,d} \end{bmatrix} \quad (11)$$

其中, X 为麻雀个体, n 为麻雀数量, d 表示要优化的变量的维度。

个体的适应度值可由如下矩阵表示:

$$F_X = \begin{bmatrix} f([X_{1,1} & X_{1,2} & \cdots & \cdots & X_{1,d}]) \\ f([X_{2,1} & X_{2,2} & \cdots & \cdots & X_{2,d}]) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ f([X_{n,1} & X_{n,2} & \cdots & \cdots & X_{n,d}]) \end{bmatrix} \quad (12)$$

在SSA算法中,更满足适应度函数的个体会优先获得食物。因此通过对个体适应度的值进行排序,划分角色,利用下式对麻雀进行位置更新^[14]。

探索者的位置计算公式为:

$$X_{j+1}^{i,d} = \begin{cases} X_j^{i,d} \exp\left(\frac{-i}{\alpha T}\right), & R_2 < S \\ X_j^{i,d} + QL, & R_2 \geq S \end{cases} \quad (13)$$

式中: $X_j^{i,d}$ 为种群中第 j 次迭代时,第 i 只麻雀的第 d 个维度上的值; T 为最大迭代次数; $\alpha \in (0, 1]$ 是一个随机数; R_2 和 S 分别表示警报值和安全阈值。 Q 为满足正态分布的随机数, L 是一个 $1 \times d$ 的矩阵,里面元素全为1。

追随者位置计算公式为:

$$X_{j+1}^{i,d} = \begin{cases} Q \exp\left(\frac{X_{p(j+1)}^d - X_j^{i,d}}{i^2}\right), & \text{if } i > \frac{n}{2} \\ X_{p(j+1)}^d + |X_{j+1}^d - X_{p(j+1)}^d| \cdot A^+ \cdot L, & \text{其他} \end{cases} \quad (14)$$

式中: X_{wj}^d 为第 j 次迭代时当前全局的最差位置; $X_{p(j+1)}^d$ 为第 $j+1$ 次迭代时当前全局的最佳位置; A^+ 为元素赋值随机1或-1、大小为 $1 \times d$ 的矩阵。

同时麻雀种群还存在警戒机制,会有一部分麻雀能够感知到可能的捕食危险进而调整位置^[15],警戒者的位置计算公式为:

$$X_{j+1}^{i,d} = \begin{cases} X_{bj}^d + \beta(X_j^{i,d} - X_{bj}^d), & F_i > F_g \\ X_j^{i,d} + K \left(\frac{X_j^{i,d} - X_{wj}^d}{|F_i - F_w| + \delta} \right), & F_i = F_g \end{cases} \quad (15)$$

式中: X_{bj}^d 为当前全局最佳位置; β 为步长控制参数; F_i 为第 i 只个体的适应度; F_g 和 F_w 分别为集群当前最优和最差适应度值; K 为麻雀移动方向,取值为 $[-1, 1]$ 的随机数; δ 为接近于0的一个极小常

数,防止分母为0^[16]。麻雀搜索算法步骤流程图如图3所示。

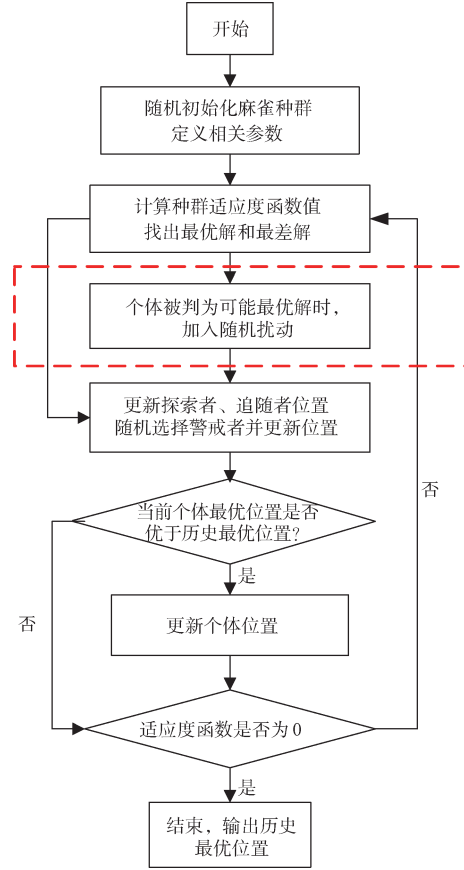


图3 麻雀搜索算法流程图

Fig 3 Sparrow search algorithm flowchart

在判断个体是否为最优解时,创新性地加入随机扰动,有助于打破算法的收敛稳定性,避免在程序运行初期就找到局部最优解的情况发生,这样可以最大程度增强算法的全局探索性。

3 仿真分析

通过MATLAB搭建TTE网络系统,设置16个网络节点,其中包括4个交换机和12个网络终端,组成的网络拓扑结构如图4所示。设置整个系统传输80条TT帧,每条TT帧的传输路径由MATLAB随机生成,在此基础上求解静态调度表。

如图5所示是一个集群周期内TT帧静态调度表的求解结果。 X 轴代表“时间槽”序号, Y 轴代表数据链路序号,每一个点即为一个TT帧。由图中可以看出每条链路上TT帧的传输情况。

使用粒子群算法与SSA算法作对比仿真,如图6所示是基于粒子群算法的TTE静态调度表求解

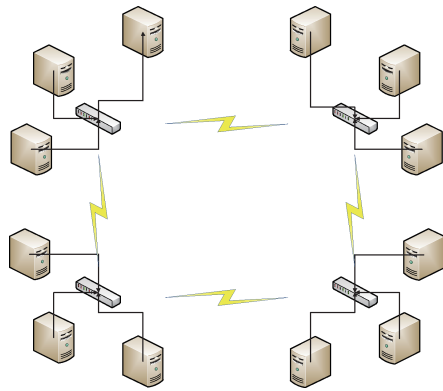


图 4 TTE 网络拓扑结构

Fig 4 TTE network topology

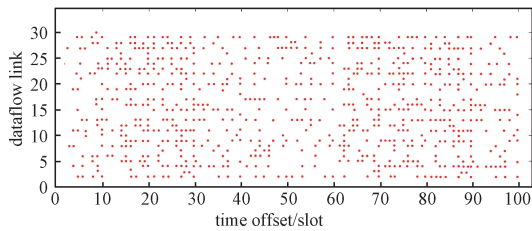


图 5 TT 帧调度仿真结果

Fig 5 The scheduling simulation results of TT frame

结果。

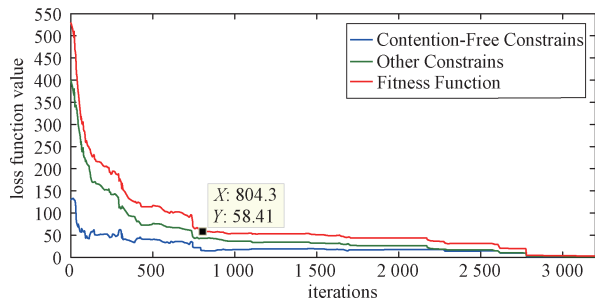


图 6 粒子群算法仿真结果

Fig 6 The simulation results of particle swarm optimization

其中红线表示总适应度函数 pf , 蓝线表示无竞争约束下的损失函数 pf_1 , 绿线是中继节点约束下的损失函数 pf_2 和端到端传输约束下的损失函数 pf_3 的和。该算法与 SSA 算法原理类似, 但由于粒子只会向个体最优方向移动, 因此容易陷入局部最优解^[17]。由图中可以看出, 在迭代 800 次后速度变慢, 在约 2 800 次处收敛到 0, 此时算法结束, 静态调度表生成。

本文 1.3.2 节所提出的约束条件收敛性如图 7 所示。

其中绿线表示无竞争约束下的损失函数 pf_1 , 由表 1 可知, 改进的 SSA 算法虽然在求解时间

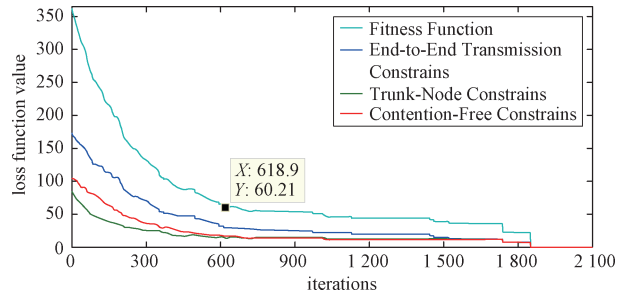


图 7 SSA 算法仿真结果

Fig 7 The simulation results of SSA algorithm

红线表示中继节点约束下的损失函数 pf_2 , 深蓝线是端到端传输约束下的损失函数 pf_3 , 浅蓝线表示总适应度函数 pf 。由图可以看出, 端到端传输约束对适应度函数的影响最大, 无竞争约束对适应度函数影响最小。明显看出, 收敛速度在迭代开始时较快, 在 600 次后变慢, 在约 1 800 次处收敛到 0, 此时算法结束, 即每一条 TT 帧均找到最优调度路径。

SSA 算法本身特性可以避免陷入局部最优, 提升全局搜索能力, 本文在算法仿真过程中对个体最佳位置增加随机扰动, 达到进一步寻找全局最优解的目的。为验证这一改进对算法收敛性的影响, 控制种群的初值相同, 在个体最佳位置分别输入加入随机扰动及不加入随机扰动的 SSA 算法, 得到如图 8 所示的收敛情况。其中红线表示不加扰动后的 SSA 算法收敛曲线, 适应度函数最终稳定在 100 左右, 蓝线表示加入扰动后的 SSA 算法收敛曲线, 大约在迭代 1 800 次后收敛到 0, 具有良好的收敛性, 且初期的下降速度较为可观。

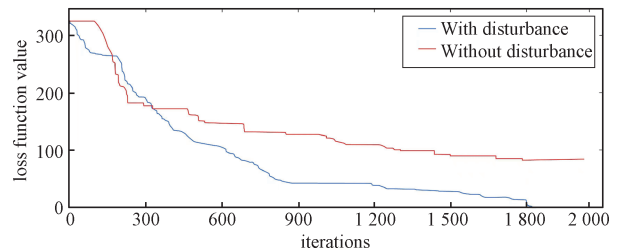


图 8 随机扰动对算法收敛性的影响

Fig 8 THE influence of random perturbation on convergence

对于粒子群算法、SSA 算法和加入扰动的改进 SSA 算法, 设置相同的传输消息数 30 条和最大迭代次数 1 000 次, 单独运行 20 次, 结果取均值, 得到如下对比结果如表 1 所示。

上较长, 但在平均值和标准差上优势明显, 说明

表1 各类算法结果对比
Table 1 Comparison of results of various algorithms

算法	平均值/标准差	求解效率/s	链路时间资源占用率	CPU占用率
粒子群算法	4.58E+02/2.95E+02	2.59	7.2%	92.4%
SSA算法	7.32E-16/5.54E-15	2.83	6.8%	92.7%
改进SSA算法	6.18E-56/0.00E+00	>100	3.1%	95.2%

具有良好的稳定性和精确度。在链路时间资源占用率这一关键指标中，粒子群算法和SSA算法资源消耗基本相同，而改进后的算法降低了3.6%，体现了加入扰动的SSA算法的优越性。

4 结束语

本文对TTE静态调度表的求解方法进行了数学建模，在此基础上，选择了操作较为简单且适合于工程应用的SSA算法进行优化。结果表明：本文所实现的基于SSA算法的TTE静态调度表求解方法具有良好的收敛性，大约迭代到1800次后收敛到0，且初期收敛速度较快，若在个体最优位置加入扰动，可以进一步提升收敛效果，具有很好的应用价值。

今后联系工程实践，本文尚有可以继续研究的方向，比如RC帧和BE帧的传输也应与TT帧一样预先求解调度方案，以及工程上应用层的约束条件如何转化为数学模型进行求解等。

参考文献

- [1] 邹伟, 郝现伟, 李菲, 等. 新一代航天电子系统现场总线及其应用[J]. 遥测遥控, 2021, 42(3): 34-39. DOI: 10.13435/j.cnki.ttc.003136.
ZOU Wei, HAO Xianwei, LI Fei, et al. A new generation of fieldbus in aerospace electronic system and its application[J]. Journal of Telemetry, Tracking and Command, 2021, 42(3): 34-39.
- [2] 刘锡鹏, 陈寅. 基于SMT的机组排班问题优化求解[J]. 计算机系统应用, 2021, 30(12): 279-287. DOI:10.15888/j.cnki.csa.008181.
LIU Xipeng, CHEN Yin. Optimal solution to crew scheduling problem based on SMT[J]. Computer Systems & Applications, 2021, 30(12): 279-287. DOI: 10.15888/j.cnki.csa.008181.
- [3] 商行. TTE网络调度表生成方法研究[D]. 西安: 西安电子科技大学, 2019. DOI: 10.27389/d.cnki.gxadu.2019.000263.
- [4] 张春韵, 邹德旋, 沈鑫. 改进的粒子群算法在电力经济调度中的应用[J]. 制造业自动化, 2021, 43(1): 53-57, 64. ZHANG Chunyun, ZOU Dexuan, SHEN Xin. Application of improved particle swarm optimization algorithm in power economic dispatching[J]. Manufacturing Automation, 2021, 43(1): 53-57, 64.
- [5] 范雅男, 逢焕利. 求解柔性流水车间调度问题的粒子群优化算法[J]. 长春工业大学学报, 2022, 43(3): 258-264. DOI:10.15923/j.cnki.cn22-1382/t.2022.3.11.
FAN Yanan, PANG Huanli. Particle swarm optimization algorithm for flexible flow shop scheduling problem[J]. Journal of Changchun University of Technology, 2022, 43(3): 258-264. DOI:10.15923/j.cnki.cn22-1382/t.2022.3.11.
- [6] 陈春燕, 王红春, 王小辉. 基于强化学习的TTE网络调度规划方法设计与实现[J]. 实验技术与管理, 2023, 40(4): 52-61, 74. DOI:10.16791/j.cnki.sjg.2023.04.007.
CHEN Chunyan, WANG Hongchun, WANG Xiaohui. Design and implementation of reinforcement learning-based scheduling planning method for TTE network[J]. Experimental Technology and Management, 2023, 40(4): 52-61, 74. DOI:10.16791/j.cnki.sjg.2023.04.007.
- [7] 刘齐坚, 王韦刚, 高鹏程. 基于GSLF-SSA的异构多核处理器任务调度[J]. 计算机技术与发展, 2024, 34(7): 48-54. DOI:10.20165/j.cnki.ISSN1673-629X.2024.0121.
LIU Qijian, WANG Weigang, GAO Pengcheng. Task scheduling for heterogeneous multi-core processors based on GSLF-SSA[J]. Computer Technology and Development, 2024, 34(7): 48-54. DOI:10.20165/j.cnki.ISSN1673-629X.2024.0121.
- [8] 岳润雨, 张晓林, 张展. 基于Stateflow的TTE通信网络仿真和性能验证[J]. 遥测遥控, 2016, 37(2): 61-66. DOI: 10.13435/j.cnki.ttc.002773.
YUE Runyu, ZHANG Xiaolin, ZHANG Zhan. Simulation and performance verification of time-triggered ethernet based on Statrflow[J]. Journal of Telemetry, Tracking and Command, 2016, 37(2): 61-66. DOI:10.13435/j.cnki.

- ttc.002773.
- [9] 黄进建. 改进型 TTE 交换机 TT 平面的设计与实现[D]. 西安: 西安电子科技大学, 2022. DOI:10.27389/d.cnki.gxadu.2022.002253.
- [10] 王家兴. 时间触发以太网关键技术研究与设计[D]. 杭州: 浙江大学, 2019.
- [11] 张景斌, 刘炯, 申普兵. 一种基于等长时隙划分双层卫星网络路由算法[J]. 中国空间科学技术, 2015, 35(3): 17-25.
ZHANG Jingbin, LIU Jiong, SHEN Pubing. A routing algorithm for double-layer satellite network based on equal-length time slot division[J]. Chinese Space Science and Technology, 2015, 35(3): 17-25.
- [12] 薛建凯. 一种新型的群智能优化技术的研究与应用[D]. 上海: 东华大学, 2020. DOI: 10.27012/d.cnki.gdhuu.2020.000178.
- [13] 吴满毅, 徐良骥, 张坤. 基于 SSA-BP 神经网络的概率积分法预计参数求取研究[J]. 金属矿山, 2022(8): 182-189. DOI:10.19614/j.cnki.jsks.202208025.
WU Manyi, XU Liangji, ZHANG Kun. Estimated parameter extraction research of probabilistic integration method based on SSA-BP neural network[J]. Metal Mine, 2022(8): 182-189. DOI: 10.19614/j.cnki.jsks.202208025.
- [14] 尹德鑫, 张达敏, 蔡朋宸, 等. 改进的麻雀搜索优化算法及其应用[J]. 计算机工程与科学, 2022, 44(10): 1844-1851.
YIN Dexin, ZHANG Damin, CAI Pengchen, et al. An improved sparrow search optimization algorithm and its application[J]. Computer Engineering & Science, 2022, 44(10): 1844-1851.
- [15] 王振东, 汪嘉宝, 李大海. 一种增强型麻雀搜索算法的无线传感器网络覆盖优化研究[J]. 传感技术学报, 2021, 34(6): 818-828.
WANG Zhendong, WANG Jiabao, LI Dahai. Study on WSN optimization coverage of an enhanced sparrow search algorithm[J]. Chinese Journal of Sensors and Actuators, 2021, 34(6): 818-828.
- [16] 刘元杰, 崔建勇, 董文, 等. 基于 ISSA-SVR 算法的近海典型水质参数遥感反演[J]. 遥测遥控, 2024, 45(3): 81-90.
LIU Yuanjie, CUI Jianyong, DONG Wen, et al. Remote sensing retrieval of coastal water quality parameters based on ISSA-SVR method[J]. Journal of Telemetry, Tracking and Command, 2024, 45(3): 81-90.
- [17] 谢天壹, 韩倩倩, 苗洪利, 等. 基于粒子群算法的相控阵海浪波谱仪天线增益校正方法[J]. 遥测遥控, 2024, 45(1): 116-125. DOI:10.12347/j.ycyk.20231103001.
XIE Tianyi, HAN Qianqian, MIAO Hongli, et al. Antenna gain calibration method for phased array wave spectrometer based on particle swarm algorithm[J]. Journal of Telemetry, Tracking and Command, 2024, 45(1): 116-125. DOI:10.12347/j.ycyk.20231103001.

[作者简介]

- 刘 冉 1997年生, 硕士研究生。
卢满宏 1968年生, 博士, 研究员。
张显辉 1976年生, 硕士, 研究员。
肖佳洛 1986年生, 硕士, 高级工程师。
杨文涛 1987年生, 硕士, 高级工程师。

(本文编辑: 潘三英)

(英文编辑: 赵尹默)