

# TS\_scheduler 类

## 1 类概述

TS\_scheduler 类，继承 dynamic\_scheduler 类。通过禁忌搜索算法对 SFC 进行部署。

## 2 类属性

名称	描述	类型	默认值
__records	部署记录	{}	{}
tabu_list	禁忌表	[]	[]
tabu_length	禁忌长度限制	int	10
interaction_count	迭代次数	int	50
stop_condition	停止条件	int	5
max_deploy_record	单前最优部署记录	[]	无
max_deploy_solution	上述最优部署记录对应数字记录	int	无
global_max_deploy_record	全局最优部署成绩	int	初始解成绩
grade_list	搜索成绩表	[]	无
all_sfc_deploy_solutions	Sfc 的所有部署方案	{'sfc1':[{}], {'sfc2':[{}], ...}}	无
solutions_length	每个 sfc 的部署方案个数	{'sfc1':10, 'sfc2':5, ...}	无

## 3 类方法

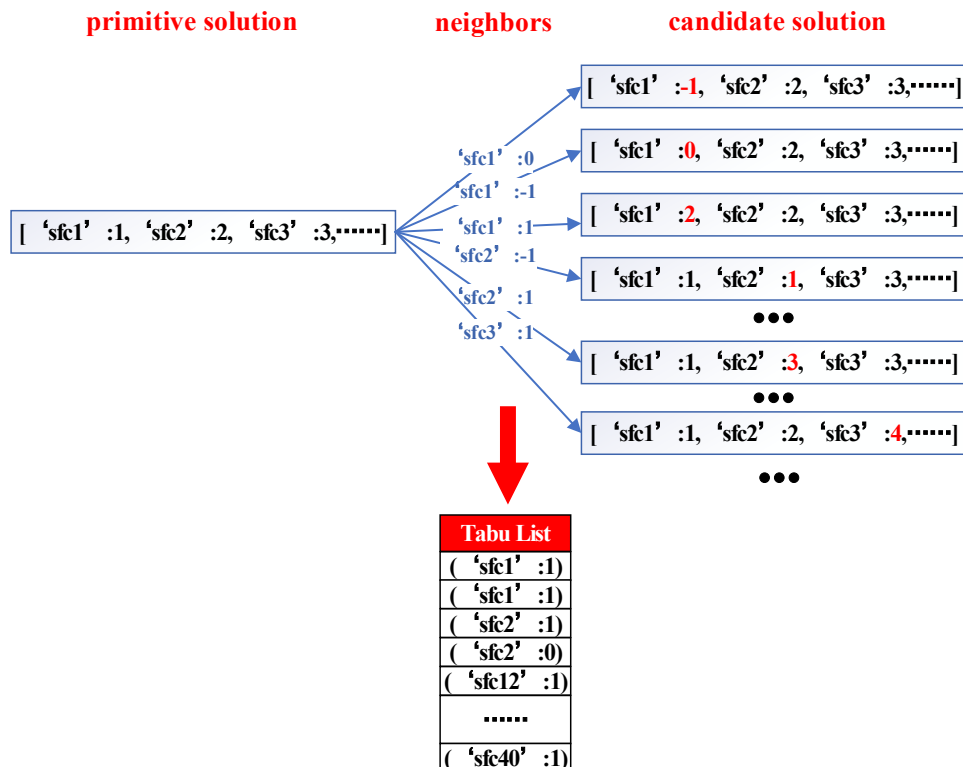
名称	描述
__init__(tabu_length=10, iteration=50, stop_condition=5)	类内部构造函数
init(init_record, network, sfcs)	初始化，提前计算一些常用的量
clear_network(network, sfcs)	清除网络
check_score(record, sfcs)	计算流量大小
deploy_sfc_by_records(sfcs, network, vnf_types, records)	通过记录执行部署操作并计算返回适应度

get_new_deploy_solution(neighbour)	获得进行邻域操作之后的新部署方案
get_last_deploy_solution (neighbour)	回到邻域操作之前的部署方案
get_neighbour(afc_id)	获得一条 sfc 的邻域
get_neighbours()	获得单前解的所有邻域
is_in_tabu_list(neighbour)	判断邻域是否在禁忌表中
calculate_fits(sfcs, network, vnf_types, neighbours)	计算邻域的适应度
single_search(network, sfcs, vnf_types)	#执行一次搜索
deploy_sfcs(network, sfcs, vnf_types, init_record)	主函数

## 4 禁忌搜索流程

### 4.1 邻域更新以及禁忌表

先找到每一条 sfc 部署的所有解方案，并以数字连续编码，其中-1 代表不部署此条 sfc。然后按照下图所示邻域操作以及禁忌表方案。



## 4.2 禁忌搜索流程

