

## ◎热点与综述◎

## 超启发算法研究进展综述

谢 毅, 侯彦娥, 陈小潘, 孔云峰

XIE Yi, HOU Yan'e, CHEN Xiaopan, KONG Yunfeng

河南大学 环境与规划学院, 河南 开封 475004

College of Environment and Planning, Henan University, Kaifeng, Henan 475004, China

XIE Yi, HOU Yan'e, CHEN Xiaopan, et al. Review of research progress of hyper-heuristic algorithms. *Computer Engineering and Applications*, 2017, 53(14): 1-8.

**Abstract:** Hyper-heuristics algorithms seek to efficiently solve the combinatorial optimization problems. It is expected to handle classes of problems rather than solving just one problem. This paper presents a systematic review of the research progress of hyper-heuristic algorithms. Based on a brief introduction to hyper-heuristic definition, structure and classification, the heuristic selection and heuristic generation approaches of hyper-heuristics are summarized. The techniques such as the machine learning methods in choosing low-level heuristics, the move acceptance criteria for neighborhood solutions, the generation of low-level heuristics, and the algorithm framework are discussed in details. Finally, the limitations and further research directions of hyper-heuristics algorithms are discussed.

**Key words:** hyper-heuristics algorithm; selection; generation

**摘 要:**超启发算法是一类新兴的优化方法,通过机器学习、算法选择、算法生成等技术求解组合优化等问题,具备跨问题领域求解的能力。针对超启发算法研究进展进行综述和讨论。首先,梳理超启发算法的定义、结构、特点和分类;其次,归纳选择式超启发算法和生成式超启发算法的研究进展及相关技术,包括选择低层启发式算法采用的学习方法,迭代计算中的移动接受策略,低层启发式算法的生成方法;最后,讨论现有超启发算法研究中存在的不足及未来的研究方向。

**关键词:**超启发算法;选择式;生成式

**文献标志码:**A **中图分类号:**TP301 **doi:**10.3778/j.issn.1002-8331.1611-0131

## 1 引言

目前,求解组合优化问题的主要方法包括精确算法和启发式算法。精确算法能够获取问题最优解,但计算时间随问题规模扩大呈指数级增长。启发式算法已广泛应用于求解各类组合优化问题,但也存在一些问题。首先,启发式算法往往被设计成求解某一专门问题,缺乏通用性;其次,对算法设计人员的能力要求高,需要设计者同时具备很高的专业背景知识和算法设计技巧;最后,实际应用中的案例千差万别,根据没有免费的午餐

理论<sup>[1]</sup>,某一算法针对某些案例求解性能良好,但针对其他案例时往往表现不佳。为解决上述问题,超启发算法(Hyper-Heuristics Algorithm)的概念被提出并迅速引起学术界关注,已有学者将其应用于排课问题、调度问题、装箱问题等组合优化问题的求解。本文总结了超启发算法的研究现状,归纳实现超启发算法的相关技术,讨论现有超启发算法研究中存在的不足和未来的研究方向,为超启发算法的进一步研究和应用提供参考。

**基金项目:**国家自然科学基金(No.41401461);河南大学科研基金(No.2015YBZR024)。

**作者简介:**谢毅(1983—),男,博士研究生,讲师,主要研究方向为空间优化, E-mail: xieyi@henu.edu.cn;侯彦娥(1980—),女,博士,讲师,主要研究方向为人工智能、空间优化;陈小潘(1982—),男,博士,讲师,主要研究方向为空间优化;孔云峰(1967—),男,博士,教授,主要研究方向为GIS工程分析与设计、空间优化。

**收稿日期:**2016-11-08 **修回日期:**2017-05-09 **文章编号:**1002-8331(2017)14-0001-08

2 超启发算法结构、特点及分类

2.1 超启发算法结构

Cowling 最早提出了超启发算法<sup>[2]</sup>并用于求解调度问题<sup>[3-6]</sup>,作者将超启发算法描述为“寻找启发式算法的启发式算法”。Burke 对超启发算法进行了更精确的定义:“超启发算法提供了一种高层启发式方法,通过管理或操纵一系列低层启发式算法,用于求解各种组合优化问题<sup>[7]</sup>”。

典型的超启发算法在逻辑结构上由控制域和问题域两个部分组成,如图 1 所示。问题域中包含由领域专家设计的问题描述、基本函数、评价函数以及若干低层启发式算法(Low-Level Heuristics, LLH);控制域中的高层控制策略(High-Level Strategy, HLS)由超启发算法专家进行设计,包含了如何利用低层启发式算法构造可行解或者提升解质量的方法。问题域及控制域之间是领域屏蔽,定义了两层结构之间进行信息传递的标准接口。

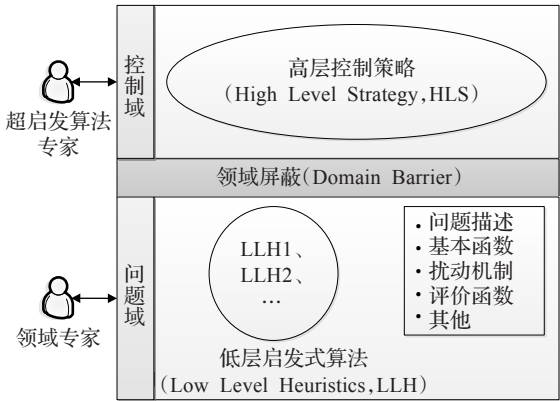


图1 超启发算法逻辑结构

2.2 超启发算法特点

由超启发算法的逻辑结构可以看出其具备以下特点。  
(1)超启发算法的搜索对象为低层启发式算法,因此其抽象程度高于传统启发式算法。

(2)超启发算法专家较少需要或不需要具备问题域的专业知识。超启发算法包括控制域和问题域两层结构,分别由超启发算法专家与问题领域专家设计。理想情况下,控制域只需了解问题域中的低层启发式算法的数量以及评价函数,就能进行问题的求解。

(3)超启发算法具备通用性。当超启发算法求解新问题领域时,通过领域屏蔽中的标准接口,只需在问题域中设计与新问题相关的低层启发式算法、问题描述及评价函数等内容,就能将一种高层控制策略应用于新问题领域的求解。

可以看出,超启发算法与传统启发式算法存在着很多不同,如表 1 所示。

超启发式算法的目标是将智能计算技术更快地推广到更多的领域,同时降低启发式算法设计的难度,将

表 1 超启发算法与传统启发式算法异同

	传统启发式算法	超启发算法
搜索空间	问题解空间	低层启发式算法集合
专业知识	设计人员需要同时具备智能计算知识和问题领域知识	控制域设计人员较少或不需要具备问题领域知识
通用性	面对新问题,一般需要重新设计	高层控制策略可以应用于不同的问题领域

领域专家和智能计算专家的定位区分开来。智能计算专家主要关注超启发式算法的高层控制策略,而领域专家侧重于关注低层启发式算法和问题的评价函数等<sup>[8]</sup>。

2.3 超启发算法分类

超启发算法与传统启发式算法本质上都是通过不断搜索邻域空间以提高解的质量。因此传统启发式算法的理论和方法也能应用于超启发算法中,其分类方法也与传统启发式算法类似,通常将搜索邻域空间采取的方法作为主要的分类依据。

Burke<sup>[9]</sup>依据高层控制策略在选择低层启发式算法时反馈信息的来源和搜索空间的性质进行分类,如图 2 所示。

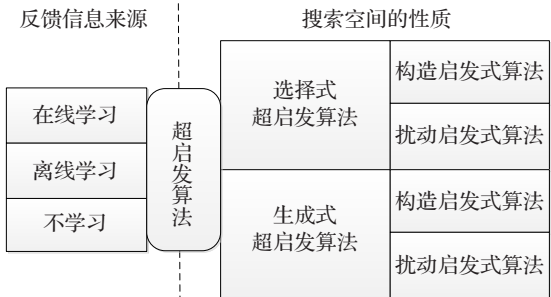


图2 Burke提出的超启发算法分类方法

依据超启发算法选择低层启发式算法时采用的反馈信息来源不同,可以将超启发算法分为不学习(No Learning)、离线学习(Offline Learning)和在线学习(Online Learning)三种类型。不学习机制在计算时通过随机方式确定低层启发式算法的组合方式和执行顺序;离线学习机制通过样本数据训练获得低层启发式算法与问题特征之间的关系,求解实际案例时根据这些关系确定低层启发式算法的执行顺序。在线学习机制在每次迭代中利用低层启发式算法的历史信息确定下一个迭代过程中使用的低层启发式算法。

根据超启发算法搜索空间的性质,可以将超启发算法分为选择式超启发算法(Heuristic Selection)和生成式超启发算法(Heuristic Generation)。选择式超启发算法从一组预先定义好的低层启发式算法进行选择;生成式超启发算法提供的是启发式算法组件,在计算过程中通过这些组件生成新启发式算法进行求解。低层启发式算法根据其特性可分为构造启发式算法(Construction heuristics)和扰动启发式算法(Perturbation heuristics)。构造启发式算法的目的是从空解开始逐步建立一个可

行解,扰动启发式方法通过搜索解的邻域空间,尝试找到质量更好的解。

3 选择式超启发算法

选择式超启发算法提供了一组预先定义的低层启发式算法集合,在求解过程中,高层控制策略从低层启发式算法集合中选取一个低层启发式算法序列构造可行解或者改进当前解,根据低层启发式算法的性质不同,可以将选择式超启发算法分为基于构造的选择式超启发算法和基于扰动的选择式超启发算法。

3.1 低层启发式算法的选择方法

在选择式超启发算法中,研究者主要关注低层启发式算法的选择方法。根据反馈信息来源不同,选择方法可以分为不学习、离线学习和在线学习三类。

(1)不学习机制设计最为简单,在计算时通过随机方式确定低层启发式算法的组合方式和执行顺序,不需要反馈低层启发式算法运行时的信息。Ahmadi<sup>[10]</sup>和Sabar<sup>[11]</sup>等学者采用随机策略选择低层启发式算法,用于求解教育排课问题。

(2)离线学习机制通常采用学习系统训练样本数据,通过反复的评价测试以优化求解模型。求解真实数据时,可根据问题特征找到相应的求解模型,反馈信息在求解实际案例之前已经确定,其原理如图3所示。例如在Ross and Schulenburg<sup>[12]</sup>求解一维装箱问题的研究中,作者设计了对当前问题的简单描述,包括待装箱物品的数量以及各个尺寸所占比例(巨大,大,中,小)。作者预先对低层启发式算法在不同问题描述下的表现进行分类训练。在求解实际问题时,高层策略根据实际问题描述和训练结果找到合适的低层启发式算法组合对问题进行求解。

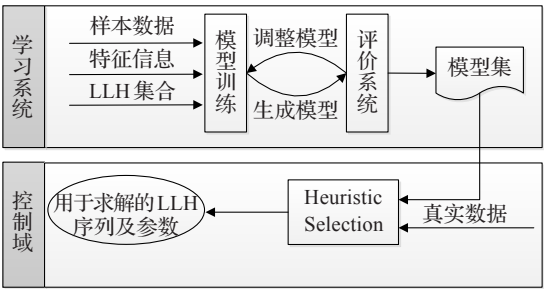


图3 离线学习原理示意图

(3)在线学习机制中,高层控制策略每次选择都要获取低层启发式算法的运行信息。在线学习机制可分为基于元启发算法的选择方法、基于强化学习的选择方法和基于选择函数的选择方法。

基于元启发算法的选择方法通常使用禁忌算法和进化算法。禁忌算法将上个迭代过程中使用的低层启发式算法加入禁忌表中,例如在Burke<sup>[13-14]</sup>的研究中采用了禁忌算法求解教育排课问题。进化算法则将低层启

发式算法表示为染色体,每次迭代通过交叉、变异等操作选择新的低层启发式算法。Vázquez-Rodríguez<sup>[15-16]</sup>和Pillay and Banzhaf<sup>[17]</sup>的研究中使用了遗传算法选择低层启发式算法。

基于强化学习选择方法使用在线得分(Online Score)来评价低层启发式算法在计算过程中的表现,高层策略再依据分值决定在迭代过程中采用哪些低层启发式算法改进当前解。其原理如图4所示。

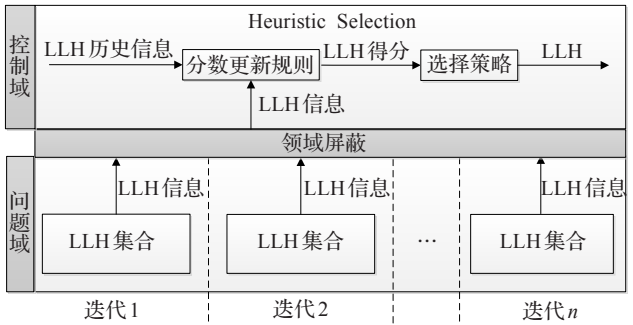


图4 强化学习原理示意图

计算在线分数的方法一般包含4个部分:初始得分、记忆长度、分数更新规则以及基于得分的算法选择策略。所有的低层启发式算法都有一个初始得分,在大部分的方法中,低层启发式算法的初始得分都一样。记忆长度保存了每个低层启发式算法在之前阶段的表现,记忆长度越长,低层启发式算法的历史信息保存得越久,反之则越短。分数更新规则一般采用强化学习方法(Reinforcement Learning),当低层启发式算法改进当前解时,就给这个低层启发式算法进行加分,反之则扣除一定分数。根据得分,超启发算法通过贪心法(Greedy)和轮盘法(Roulette-wheel)选择低层启发式算法。表2列出了这两种方法的原理。

表2 利用得分机制选择低层启发式算法方法

名称	原理
贪心选择策略	总是选择得分最高的低层启发式算法
轮盘法	每个低层启发式算法都有一定的概率被选中,被选中概率与其得分成正比

基于选择函数(Choice Function)的方法也是通过计算得分选择低层启发式算法,但计算分数的方法与强化学习方法不同。其原理是设计一个多项式函数,函数中的项代表低层启发式算法在迭代过程中的表现信息,例如使解变好的次数、使解变差的次数、运算时间、最近调用时间等,不同项的权重不同,最终依据函数计算每个低层启发式算法的得分并进行选择。

如何选择低层启发式算法是选择式超启发算法研究的核心,不学习、离线学习和在线学习三种选择机制特点不同。采用随机的方式选择低层启发式算法,虽然设计简单,但其收敛性差,在问题规模较大时很难获取高质量的解。采用离线学习机制的超启发算法通常能



够较快地获取高质量的解,缺点是需要额外的学习系统训练样本数据,设计与实现难度较大,此外,样本数据训练是否充分、特征信息的设定也对求解质量有直接影响。采用在线学习机制的超启发算法设计难度适中,同时也能获得很好的解,但是在线学习机制仅仅利用了计算过程中的信息进行策略改进,忽略了大量有用的问题领域信息,因而收敛速度比较慢。

3.2 基于构造的选择式超启发算法

基于构造的选择式超启发算法目的是构建问题可行解。该方法从一个空解开始,由高层控制策略从一组预先提供的低层构造启发式算法中(特定问题领域)选择算法逐步建立问题的可行解。算法执行过程中,当成功构建一个可行解后,算法即终止。基于构造的选择式超启发算法在教育排课问题(Education scheduling)、生产调度问题(Production scheduling)、装箱问题(Pin packing)等问题领域得到了应用。表3列出了一些相关文献。

表3 基于构造的选择式超启发算法的相关文献

引用文献	反馈信息来源
Ahmadi <sup>[10]</sup>	NO
Sabar <sup>[11]</sup>	NO
Ross and Schulenburg <sup>[12]</sup>	Offline
Cano-Belmán <sup>[18]</sup>	OffLine
L <sub>i</sub> <sup>[19]</sup>	OffLine
Vázquez-Rodríguez <sup>[15-16]</sup>	OnLine
Pillay and Banzhaf <sup>[17]</sup>	OnLine
Burke <sup>[13-14]</sup>	OnLine

注:OnLine=在线学习,OffLine=离线学习,NO=不学习

研究结果表明,基于构造的选择式超启发算法能够快速获取问题的可行解,并且在小规模案例时也能获取质量不错的解,缺点是在求解大规模案例时解的质量较差。

3.3 基于扰动的选择式超启发算法

基于扰动的选择式超启发算法利用构造启发式算法或者其他方法构造初始解,然后通过迭代的方式不断尝试改善当前解直到满足终止条件,例如迭代次数或者执行时间。

根据超启发算法同时处理解的数量,可以将基于扰动的超启发算法分为单点搜索和多点搜索两种方法<sup>[20]</sup>。单点搜索方法在同一时刻处理一个解,多点搜索方法允许同时处理多个解。

3.3.1 单点搜索方法

Bilgin 和 Özcan<sup>[21-22]</sup>认为采用单点搜索方法的超启发算法包含两个组件:选择低层启发式算法的方法(Heuristic Selection Method)和移动接受方法(Move-acceptance Method)。该类超启发算法的结构如图5所示,其中  $S_{current}$  代表当前解,  $S_{out}$  代表改进后的解。

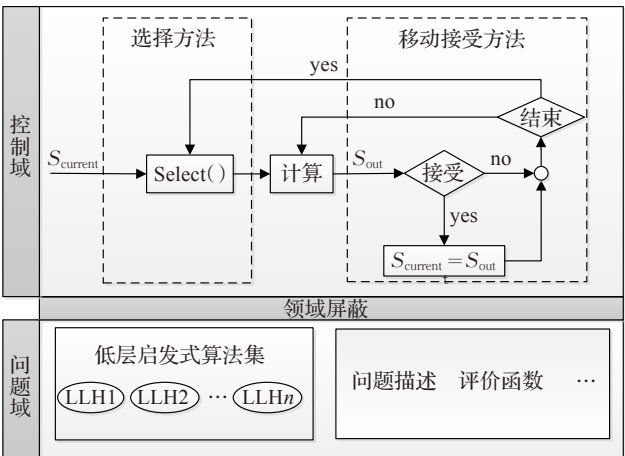


图5 基于扰动的选择式超启发算法结构图

选择方法决定了在一个迭代过程选择哪些低层启发式算法用于改进当前解,采用的方法在3.1节已经进行了介绍。

移动接受方法决定了迭代过程中产生的新解能否被接受作为当前解。文献将其分为确定性接受策略(Deterministic)和非确定性接受策略(Non-Deterministic)两种类型。确定性接受策略根据当前解以及改进解的状态,总是做出相同的决定,例如全部接受(All Move)或只接受优化解(Only Improvement),非确定性接受策略在输入相同的情况下,可能做出不同的决定。非确定性接受策略需要时间等额外的参数辅助决策,典型的非确定接受策略包括模拟退火算法(Simulated Annealing)、门槛算法(Threshold Accepting)、大洪水算法(Great Deluge)、延迟接受算法(Late Acceptance)、蒙特卡罗方法(Monte Carlo Method)、记录更新法(Record-to-Record Travel)、带阈值限制的迭代接受方法(Iteration Limited Threshold Accepting, ILTA)等。各类移动接受方法的原理如表4所示。

邻域搜索类算法的性能取决于邻域搜索多样性(Diversification)和集中性(Intensification)之间的平衡机制。确定性接受策略收敛性好,但容易陷入局部最优解,非确定性接受策略在某些情况下接受差解,能够跳出局部最优,扩大邻域搜索范围。因此,采用非确定接受策略的超启发算法往往能够得到质量更好的解<sup>[20]</sup>。

低层启发式算法选择方法和移动接受策略构成了一个更高层次的模块,当模块中的方法被替换就构成了一个新的超启发算法。许多研究者在超启发算法框架中尝试组合不同的选择方法和移动接受方法。表5列出一些基于扰动的选择式超启发算法的相关文献,并说明其采用的反馈信息来源、移动接受方法和同时改进解的数量。

使用单点搜索方法的超启发算法设计难度适中,在大规模案例中也能获取质量不错的解,因此目前的研究比较充分。

表4 常见移动接受方法原理

	方法名	原理
确定性	全部接受方法	接受任何解
	只接受优化解	接受比当前最优解更好的解
	模拟退火法	通过初始温度及冷却系数在每次迭代过程中以一定概率接受差解
非确定性	门槛算法	设置门槛值,如果当前解与该值之和大于当前最优解,则接受该解
	大洪水算法	设置了初始水位(Level)以及下降速度(Speed),在每次迭代过程中,产生的新解与当前解的距离如果小于Level,则接受新解,Level按照Speed下降
	延迟接受算法	定义了一个长度为L的表记录之前L次迭代中解的值,新解每次与表记录中之前的结果进行比较,如能提高解的质量,接受该解并更新表中相应的记录
	蒙特卡罗算法	原理与模拟退火方法类似,不同的是其不包含温度参数,而是使用一个冷却时间表
	记录更新算法	设置一个Record记录当前最优解以及一个偏差系数如果当前解的值优于Record,则将Record的值设为当前解,若当前解变差并且当前解的值小于Record*偏差系数时,则接受该差解
	带阈值限制的迭代接受算法	一个差解同时满足以下两个条件就允许被接受:经过一定次数的计算后当前解仍然没有改进
		和该差解与当前最优解的偏差在允许的范围

表5 基于扰动的选择式超启发算法文献

引用文献	反馈信息	移动接受	处理解的
	来源	方法	数量
Thabtah and Cowling <sup>[23]</sup>	OffLine	D	S
Cowling <sup>[5]</sup>	OnLine	D	S
Cowling <sup>[2]</sup>	OnLine	D	S
Nareyek <sup>[24]</sup>	OnLine	D	S
Burke <sup>[25]</sup>	OnLine	D	S
Ayob and Kendall <sup>[26]</sup>	NO	ND	S
Kendall and Mohamad <sup>[27]</sup>	NO	ND	S
Kendall and Mohamad <sup>[28]</sup>	NO	ND	S
Bai and Kendall <sup>[29]</sup>	NO	ND	S
Burke and Bykov <sup>[30]</sup>	NO	ND	S
Dowsland <sup>[31]</sup>	OnLine	ND	S
Bai <sup>[32]</sup>	OnLine	ND	S
Pisinger and Ropke <sup>[33]</sup>	OnLine	ND	S
Bhanu and Gopalan <sup>[34]</sup>	OnLine	ND	S
Misur <sup>[35]</sup>	OnLine	ND	S
Misur <sup>[36]</sup>	OnLine	ND	S
Sabar <sup>[37]</sup>	OnLine	ND	S
Cowling <sup>[4]</sup>	OnLine	D	M
Burke <sup>[38]</sup>	OnLine	D	M
Chen <sup>[39]</sup>	OnLine	D	M
Ren <sup>[40]</sup>	OnLine	D	M
Grobler <sup>[41]</sup>	OnLine	D	M
Sabar <sup>[42]</sup>	OnLine	ND,D	M
Misur <sup>[43]</sup>	OnLine	ND	S
Ping-Che <sup>[44]</sup>	OnLine	ND	S
Asta <sup>[45]</sup>	OffLine	ND	S
Kheiri <sup>[46]</sup>	OnLine	ND	S

注:OnLine=在线学习,OffLine=离线学习,NO=不学习,D=确定性接受,ND=非确定性接受,S=单点搜索,M=多点搜索

3.3.2 多点搜索方法

多点搜索方法能够同时搜索改进多个解,其原理类似于蚁群算法(CA)、基因算法(GA)等种群类元启发算法,将染色体表示为一段整数序列,其中每个整数代表一个低层启发式算法,通过设置交叉概率和变异概率,

在每个迭代阶段生成新的染色体改进当前解。  
多点搜索方法具有良好的全局搜索能力,不会陷入局部最优;并且利用它的内在并行性,可以方便地进行分布式计算,加快求解速度。但是多点搜索算法的局部搜索能力较差,计算后期搜索效率较低。

4 生成式超启发算法

生成式超启发算法利用现有组件生成新的启发式算法,而不是预先定义的完整的启发式算法。文献中通常采用遗传编程(Genetic Programming)技术进行求解。其原理是将一些规则表达为具有交叉、变异功能的树形结构或者字符串,然后利用这些规则和启发式算法组件生成新的启发式算法进行求解。  
在Bader<sup>[47]</sup>的研究中,作者利用遗传编程技术求解布尔逻辑的可满足性问题(SAT)。作者设计了若干求解SAT问题的组件,如图6所示。

```
start → FLIP v
v      → RANDOM 1
        | MAX_SCR 1 | MAX_SCR 1, op
        | IFV prob, v, v
        | MIN_SCR 1 | MIN_SCR 1, op
        | MAX_AGE 1 | MAX_AGE 1, op
1      → ALL | ALL_USC
        | RAND_USC | USC
        | IFL prob, 1, 1
        | SCR_Z 1 | SCR_Z 1, op
op     → TIE_RAND | TIE_AGE
        | TIE_SCR | NOT_ZERO_AGE
prob   → 20 | 40 | 50 |
        70 | 80 | 90
```

图6 求解SAT问题的启发式组件

这些组件在满足规则的前提下,能够构造出新的启发式算法,可以用树形结构来表示,树中的每个节点代表一个组件或者参数值,如图7所示。在求解过程中,允许树中的节点发生交叉和变异,便能构造出不同的启发式算法用于求解。  
在其他研究中,Ho和Tay<sup>[48]</sup>利用遗传编程技术求解车间机器调度问题。Dimopoulos和Zalzala<sup>[49]</sup>研究了单机作业调度问题。Fukunaga<sup>[50]</sup>利用决策树求解SAT问

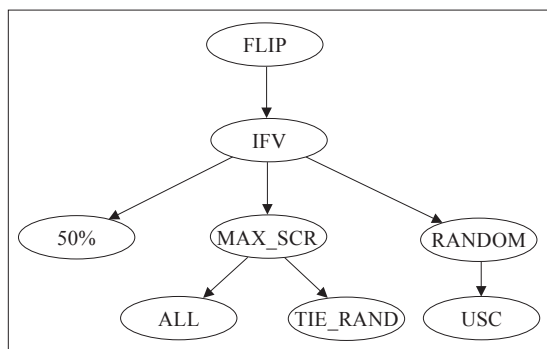


图7 树结构表示的启发式算法

题。Keller和Poli<sup>[51]</sup>利用线性遗传编程求解旅行商问题。

生成式超启发算法的搜索空间是启发式算法组件，相比选择式超启发算法，生成式超启发算法可以为问题的每一个实例生成新启发式算法。因此，生成式超启发算法更加灵活，理论上可以得到质量很高的解，但是设计和使用比较复杂。

## 5 超启发算法框架:HyFlex

Ochoa<sup>[52]</sup>公开了一个超启发算法框架HyFlex(Hyperheuristic Flexible framework)，该框架使用Java语言编写，提供通用接口解决不同问题领域的组合优化问题，在这个框架中，程序设计人员不需要关心问题领域本身，例如底层数据结构和低层启发式算法，可以集中精力设计通用的高层策略。目前，该框架提供了Binpacking、FlowShop、SAT、CVRP、PersonnelSchedule以及TSP等六类问题领域的问题描述，针对每一个问题领域提供了相应的低层启发式算法、数据结构和基准案例，研究人员只需设计选择或混合这些低层启发式算法的高层策略即可。作者利用HyFlex框架组织了Cross-Domain Heuristic Search Challenge(ChESC)竞赛，参赛人员设计超启发算法的高层策略，然后从时间效率、优化能力等若干方面评价高层策略的性能。

目前，已经有学者使用HyFlex框架进行超启发算法框架的研究。Burke<sup>[53]</sup>设计了基于在线学习的高层控制策略，Özcan和Kheiri<sup>[54]</sup>提出了一个多阶段的高层策略，在每个迭代周期内结合使用贪心选择方法和随机方法，虽然策略比较简单，但是取得了较好的实验结果。Misir<sup>[43]</sup>结合使用了强化学习和自适应阈值接受方法，并且设计了自适应阈值接受方法，结合低层启发式算法参数自适应等机制，取得了不错的计算机结果。Ping-Che<sup>[44]</sup>采用了自适应可变邻域搜索框架，作者将邻域搜索过程分为多样性和集中性两个阶段，并采用自适应技术调整邻域的规模。

Asta<sup>[45]</sup>提出一种离线学习算法在HyFlex中求解车辆路径问题，该算法分为两个阶段。第一个阶段，训练一些专家算法在限定的时间内求解指定实例，期间记录这些算法在深度搜索、变异时的参数值以及接受解的条

件；第二个阶段，利用训练的结果求解未知案例。Kheiri<sup>[46]</sup>使用多阶段迭代搜索算法和选择函数方法。

HyFlex为超启发算法研究者提供了新的途径。一方面，HyFlex提供了一套开源框架，包含了六种问题域，研究者可以专注于高层控制策略的设计与实现；另一方面，HyFlex使用基准案例进行测试，不仅提供了对各种高层控制策略进行横向对比的机会，也验证了超启发算法求解性能与传统启发式算法相比是有竞争力的，表明了超启发算法在具备通用性等优点的同时，也具备获取高质量解的能力。

## 6 总结与展望

超启发算法的能力和特点已经在一些文献中得到验证，实际应用潜力巨大。国外相关研究文献的数量呈现逐年递增的趋势。最近几年，智能计算领域的三大著名国际会议(GECCO、PPSN和CEC)都举办了专门针对超启发式算法的讨论会。

虽然超启发算法的研究已取得很多进展，但也存在一些需要改进或者提高的地方。

(1)实际应用研究不足。目前文献中的超启发算法研究大多用于求解教育排课、调度、装箱等约束较少的组合优化问题，对于现实世界中车辆路径问题、聚类问题等更为复杂的问题研究较少。

(2)缺少对搜索空间的理论分析。传统启发式算法通常使用适应度地貌分析(Fitness Landscape)描述搜索空间的特征，一些文献通过分析问题搜索空间的特征，设计了高效的启发式算法。超启发算法的搜索空间是低层启发式算法的集合，在能否使用类似理论提高求解效率上还缺少相关研究。

(3)持续探索类似HyFlex这类开放的超启发算法框架，借助此类框架可以对比不同高层控制策略求解同一问题领域或者同一高层控制策略求解不同问题领域时的性能，有助于加快对超启发算法的研究。

(4)关注机器学习领域的最新成果。机器学习(Machine Learning)能够重新组织已有的知识结构使之不断改善自身的性能，这点与超启发算法的学习机制需要利用低层启发式算法的历史信息非常相似，是否可以利用机器学习领域近些年发展迅速的深度学习、神经网络等技术提高超启发算法性能值得研究。

## 参考文献:

- [1] Wolpert D H, Macready W G. No free lunch theorems for optimization[J]. IEEE Transactions on Evolutionary Computation, 1997, 1: 67-82.
- [2] Cowling P, Kendall G, Soubeiga E. A hyperheuristic approach to scheduling a sales summit[C]//Proceedings of the International Conference on the Practice and The-



- ory of Automated Timetabling, 2000: 176-190.
- [3] Cowling P, Kendall G, Soubeiga E. A parameter-free hyper-heuristic for scheduling a sales summit[C]//Proceedings of the 4th Metaheuristic International Conference, 2001: 127-131.
- [4] Cowling P, Kendall G, Soubeiga E. Hyperheuristics: a robust optimisation method applied to nurse scheduling[C]//Proceedings of the 2002, Seventh International Conference on Parallel Problem Solving from Nature, 2002: 851-860.
- [5] Cowling P, Kendall G, Soubeiga E. Hyperheuristics: a tool for rapid prototyping in scheduling and optimisation[J]. Lecture Notes in Computer Science, 2002, 2279: 1-10.
- [6] Cowling P, Kendall G, Han L. An investigation of a hyper-heuristic genetic algorithm applied to a trainer scheduling problem[C]//Proceedings of the Evolutionary Computation, 2002: 1185-1190.
- [7] Burke E K, Curtois T, Hyde M, et al. Hyflex: a flexible framework for the design and analysis of hyper-heuristics[C]//Proceedings of the Multidisciplinary International Scheduling Conference (MISTA 2009), 2009: 790-797.
- [8] 江贺. 超启发式算法: 跨领域的问题求解模式[J]. 中国计算机学会通讯, 2011, 7(3): 63-70.
- [9] Burke E K, Hyde M, Kendall G, et al. A classification of hyper-heuristic approaches[M]//Handbook of metaheuristics.[S.l.]: Springer, 2010: 449-468.
- [10] Ahmadi S, Barone R, Cheng P, et al. Perturbation based variable neighbourhood search in heuristic space for examination timetabling problem[C]//Proceedings of Multidisciplinary International Scheduling: Theory and Applications (MISTA 2003), 2003: 13-16.
- [11] Sabar N R, Ayob M, Qu R, et al. A graph coloring constructive hyperheuristic for examination timetabling problems[J]. Applied Intelligence, 2011, 37(1): 1-11.
- [12] Ross P, Schulenburg S, Marín-Blázquez J G, et al. Hyper-heuristics: learning to combine simple heuristics in bin-packing problems[C]//Proceedings of the GECCO, 2002: 942-948.
- [13] Burke E, Dror M, Petrovic S, et al. Hybrid graph heuristics within a hyper-heuristic approach to exam timetabling problems[J]. Operations Research/Computer Science Interfaces, 2005, 29: 79-91.
- [14] Burke E K, Petrovic S, Qu R. Case-based heuristic selection for timetabling problems[J]. Journal of Scheduling, 2006, 9(2): 115-132.
- [15] Vázquez-Rodríguez J A, Petrovic S, Salhi A. A combined meta-heuristic with hyper-heuristic approach to the scheduling of the hybrid flow shop with sequence dependent setup times and uniform machines[C]//Proceedings of the 3rd Multidisciplinary International Conference on Scheduling: Theory and Applications, 2007: 506-513.
- [16] Vázquez-Rodríguez J A, Petrovic S. A new dispatching rule based genetic algorithm for the multi-objective job shop problem[J]. Journal of Heuristics, 2010, 16(6): 771-793.
- [17] Pillay N, Banzhaf W. A genetic programming approach to the generation of hyper-heuristics for the uncapacitated examination timetabling problem[M]. Berlin/Heidelberg: Springer, 2007.
- [18] Cano-Belmán J, Ríos-Mercado R Z, Bautista J A. Scatter search based hyper-heuristic for sequencing a mixed-model assembly line[J]. Journal of Heuristics, 2010, 16(6): 749-770.
- [19] Li J. Integrating neural network and logistic regression to underpin hyper-heuristic search[J]. Knowledge-Based Systems, 2011, 24(2): 322-330.
- [20] Burke E K, Gendreau M, Hyde M, et al. Hyper-heuristics: a survey of the state of the art[J]. Journal of the Operational Research Society, 2012, 64(12): 1695-1724.
- [21] Bilgin B, Özcan E, Korkmaz E E. An experimental study on hyper-heuristics and exam timetabling[C]//Proceedings of the International Conference on Practice & Theory of Automated Timetabling, 2006: 394-412.
- [22] Özcan E, Bilgin B, Korkmaz E E. A comprehensive analysis of hyper-heuristics[J]. Intelligent Data Analysis, 2008, 12(1): 3-23.
- [23] Thabtah F, Cowling P. Mining the data from a hyper-heuristic approach using associative classification[J]. Expert Systems with Applications, 2008, 34(2): 1093-1101.
- [24] Nareyek A. Choosing search heuristics by non-stationary reinforcement learning[M]//Metaheuristics: computer decision-making.[S.l.]: Springer, 2003: 523-544.
- [25] Burke E K, Kendall G, Soubeiga E. A tabu-search hyper-heuristic for timetabling and rostering[J]. Journal of Heuristics, 2003, 9(6): 451-470.
- [26] Ayob M, Kendall G. A monte carlo hyper-heuristic to optimise component placement sequencing for multi head placement machine[C]//Proceedings of the International Conference on Intelligent Technologies, 2003: 132-141.
- [27] Kendall G, Mohamad M. Channel assignment in cellular communication using a great deluge hyper-heuristic[C]//Proceedings of the 2004 IEEE International Conference on Network, 2004: 769-773.
- [28] Kendall G, Mohamad M. Channel assignment optimisation using a hyper-heuristic[C]//IEEE Conference on Cybernetics and Intelligent Systems, 2004: 791-796.
- [29] Bai R, Kendall G. An investigation of automated programs using a simulated annealing based hyper-heuristic[M]//Metaheuristics: progress as real problem solvers.[S.l.]:

- Springer, 2005: 87-108.
- [30] Burke E K, Bykov Y. A late acceptance strategy in hill-climbing for examination timetabling problems[C]//Proceedings of the Conference on the Practice and Theory of Automated Timetabling, 2008.
- [31] Dowsland K A, Soubeiga E, Burke E. A simulated annealing based hyperheuristic for determining shipper sizes for storage and transportation[J]. *European Journal of Operational Research*, 2007, 179(3): 759-774.
- [32] Bai R, Blazewicz J, Burke E K, et al. A simulated annealing hyper-heuristic methodology for flexible decision support[J]. *4OR: Quarterly Journal of Operations Research*, 2012, 10(1): 43-66.
- [33] Pisinger D, Ropke S. A general heuristic for vehicle routing problems[J]. *Computers & Operations Research*, 2007, 34(8): 2403-2435.
- [34] Bhanu S M S, Gopalan N. A hyper-heuristic approach for efficient resource scheduling in grid[J]. *International Journal of Computers, Communications & Control*, 2008, 3(3): 249-258.
- [35] Misir M, Wauters T, Verbeeck K, et al. A new learning hyper-heuristic for the traveling tournament problem[C]//MIC'09, 2009: 19-26.
- [36] Misir M, Verbeeck K, De Causmaecker P, et al. Hyper-heuristics with a dynamic heuristic set for the home care scheduling problem[C]//IEEE Congress on Evolutionary Computation, 2010: 1-8.
- [37] Sabar N R, Zhang X J, Song A. A math-hyper-heuristic approach for large-scale vehicle routing problems with time windows[C]//IEEE Congress on Evolutionary Computation, 2015: 830-837.
- [38] Burke E, Kendall G, Landa Silva D, et al. An ant algorithm hyperheuristic for the project presentation scheduling problem[C]//IEEE Congress on Evolutionary Computation, 2005: 2263-2270.
- [39] Chen P C, Kendall G, Berghe G V. An ant based hyper-heuristic for the travelling tournament problem[C]//Proceedings of the Computational Intelligence in Scheduling, 2007: 19-26.
- [40] Ren Z, Jiang H, Xuan J, et al. Ant based hyper heuristics with space reduction: a case study of the p-median problem[C]//Proceedings of the 11th International Conference on Parallel Problem Solving From Nature, 2010: 546-555.
- [41] Grobler J, Engelbrecht A P, Kendall G, et al. Investigating the use of local search for improving meta-hyper-heuristic performance[C]//IEEE Congress on Evolutionary Computation, 2012: 1-8.
- [42] Sabar N R, Ayob M, Kendall G, et al. Grammatical evolution hyper-heuristic for combinatorial optimization problems[J]. *IEEE Transactions on Evolutionary Computation*, 2013, 17(6): 840-861.
- [43] Misir M, Verbeeck K, Causmaecker P D, et al. An intelligent hyper-heuristic framework for chesc 2011[M]. Berlin/Heidelberg: Springer, 2012.
- [44] Hsiao P C, Chiang T C, Fu L C. A vns-based hyper-heuristic with adaptive computational budget of local search[C]//2012 IEEE Congress on Evolutionary Computation, 2012.
- [45] Asta S, Özcan E. An apprenticeship learning hyper-heuristic for vehicle routing in hyflex[C]//Proceedings of the Evolving and Autonomous Learning Systems, 2014: 65-72.
- [46] Kheiri A, Özcan E. An iterated multi-stage selection hyper-heuristic[J]. *European Journal of Operational Research*, 2015, 250(1): 77-90.
- [47] Baderelden M, Poli R. Generating sat local-search heuristics using a GP hyper-heuristic framework[C]//International Conference on Artificial Evolution (Evolution Artificielle), 2007: 37-49.
- [48] Ho N B, Tay J C. Evolving dispatching rules for solving the flexible job-shop problem[C]//IEEE Congress on Evolutionary Computation, 2005: 2848-2855.
- [49] Dimopoulos C, Zalzala A M. Investigating the use of genetic programming for a classic one-machine scheduling problem[J]. *Advances in Engineering Software*, 2001, 32(6): 489-498.
- [50] Fukunaga A S. Automated discovery of local search heuristics for satisfiability testing[J]. *Evolutionary Computation*, 2008, 16(1): 31-61.
- [51] Keller R E, Poli R. Linear genetic programming of parsimonious metaheuristics[C]//IEEE Congress on Evolutionary Computation, 2007: 4508-4515.
- [52] Ochoa G, Hyde M, Curtois T, et al. Hyflex: a benchmark framework for cross-domain heuristic search[M]. Berlin/Heidelberg: Springer, 2012.
- [53] Burke E K, Gendreau M, Ochoa G, et al. Adaptive iterated local search for cross-domain optimisation[C]//Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation, 2011: 1987-1994.
- [54] Özcan E, Kheiri A. A hyper-heuristic based on random gradient, greedy and dominance[M]//Computer and information sciences II. [S.l.]: Springer, 2011: 557-563.