

# 超启发式算法： 跨领域的问题求解模式

江 贺  
大连理工大学

关键词：超启发式算法

伴随着物联网、云计算等应用领域新技术的飞速发展，越来越多的优化组合问题涌现在计算机科学家面前。在现有的计算复杂性理论下，大多数这类问题的时间复杂度是指数级的，传统的精确算法（Exact Algorithm）已不适合求解这类问题<sup>1</sup>。因此，很多学者开始研究在较短时间内能获得近似最优解的启发式算法（Heuristic Algorithm）。

启发式算法大致可分为简单启发式算法（如爬山法、贪心法等）和元启发式算法<sup>2</sup>（如蚁群算法、遗传算法、粒子群算法等）。现有问题的求解思路在各应用领域中面临着越来越大的挑战。现有启发式算法对算法设计者有较高要求，他们既要掌握各种算法的设计技巧，又要具有足够的领域知识；传统做法是针对某一问题设计一种专门算法，并在一组测试实例上进行评价（包括最好结果、平均结果、解的方差等）。实际上，对于求解问题的具体实例，这种做法存在着不足之处。单一算法无法保证在所有问题（实例）上始终优于其他算法。由于具体实例千差万别，总有某些算法性能较差。

针对上述问题，一种被称为超启发式算法（Hyper-Heuristic Algorithm）的概念被提出，并迅速引起国际学术界的高度关注。最近两年，智能计算领域的三大著名国际会议（GECCO'09、PPSN'10和CEC'10）分别举办了专门针对超启发式算法的研讨

会（Workshop或Session）。另外，智能计算领域的两大学报《启发法（Journal of Heuristics）》和《演化计算（Evolutionary Computation）》也组织了专刊，集中介绍超启发式算法的研究进展。

## 超启发式算法的概念模型

超启发式算法可以简单阐述为“寻找启发式算法的启发式算法”，其更加严格的定义如下<sup>[2]</sup>：

**定义：**超启发式算法提供了一种高层次启发式方法，通过管理或操纵一系列低层次启发式算法（Low-Level Heuristics, LLH），以产生新的启发式算法。这些新启发式算法被用于求解各类组合优化问题。

图1给出了超启发式算法的概念模型。该模型分为两个层面：在问题域层面上，应用领域专家根据自己的背景知识，在智能计算专家协助下，提供一系列LLH和问题的定义、评估函数等信息；在高层次启发式方法层面上，智能计算专家设计高效的管理操纵机制，运用问题域所提供的LLH算法库和问题特征信息，构造出新的启发式算法。由于两个层面之间实现了领域的屏蔽，只要修改问题域的LLH和问题定义、评估函数等信息，一种超启发式算法可以方便地移植到新的问题上。从上述模型可

<sup>1</sup> 实例是指一个问题定义中各个参数具体设定后所得到的问题。

<sup>2</sup> 元启发式算法提供了一种固定的框架，通过调用从属启发式算法来求解问题。

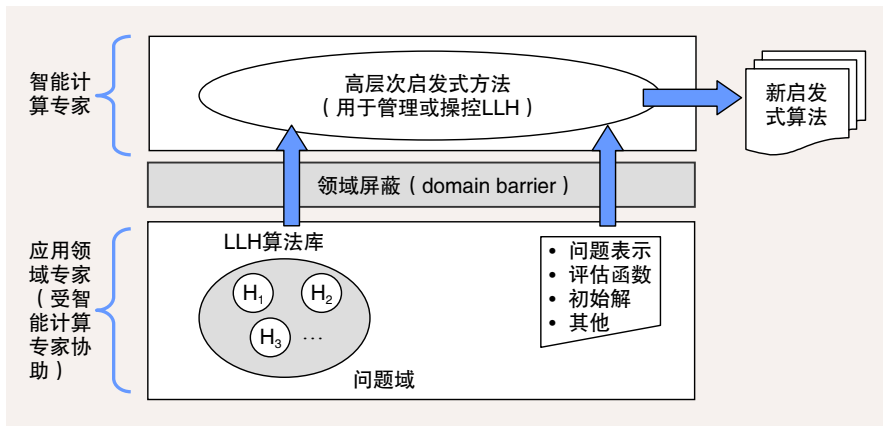


图1 超启发式算法的概念模型

表1 超启发式算法与传统启发式算法特征对比

	简单启发式算法	元启发式算法	超启发式算法
搜索空间	由实例的解构成	由实例的解构成	由启发式算法构成
问题的领域知识	需要	需要	不需要（或很少需要）
典型类别	局部搜索 爬山法 贪心法	蚁群算法 模拟退火 遗传算法 粒子群算法 可变邻域搜索	基于随机选择的超启发式算法 基于贪心策略的超启发式算法 基于元启发式算法的超启发式算法 基于学习的超启发式算法

以发现，超启发式算法具有以下特征：

1. 超启发式算法提供了一种高层次的启发式方法，它操纵管理一组LLH；
2. 超启发式算法的目标是寻找一个好的启发式算法；
3. 超启发式算法仅使用有限的领域相关信息（理想情况下，这些信息仅包括LLH数量、待求解问题的目标函数等）。

引入超启发式算法的目的不是取代智能计算专家，而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，将领域专家和智能计算专家的定位区分开来。在超启发式算法设计中，智能计算专家主要关注于高层次的启发式方法，而领域专家侧重于关注LLH和问题的目标函数等。

## 超启发式算法vs.传统启发式算法

超启发式算法与已有的启发式算法有着显著的

不同。超启发式算法运行在一个由启发式算法构成的搜索空间上，该搜索空间上的每一个顶点代表一系列LLH的组合；而传统启发式算法则是在由实例的解构成的搜索空间上工作。因此，超启发式算法的抽象程度高于传统启发式算法。另外，给定一个组合优化问题，超启发式算法可以产生不同的新启发式算法，以处理各种不同实例。表1给出了超启发式算法与传统启发式算法（简单启发式算法和元启发式算法）的一些特征对比。

## 研究现状

现有的各种超启发式算法均可以大致分为算法构造和（问题）实例求解两个阶段：前者指通过某种高层次启发式方法对一系列LLH进行组合，形成新的启发式算法；后者则是利用新启发式算法求解问题的新实例。现有的很多超启发式算法将这两个阶段融合在一起，在构造新启发式算法的同时求解问题的实例。根据算法构造阶段中高层次启发式方法的机制不同，现有的超启发式算法可以划分为以下4种类型。

### 基于随机选择的超启发式算法

该类超启发式算法是从给定的集合中随机选择LLH，组合形成新的启发式算法。这类超启发式算法在本领域研究中较早提出，其特点是结构简单、容易实现。同时，这类超启发式算法也经常被用作基准（Benchmark），以评价其他类型的超启发式算法性能。该类超启发式算法可以进一步细分为纯随机

(Pure Random)、蒙特卡罗(Monte Carlo)、带确定性接受条件的随机(Random With Deterministic Acceptance)、带延迟接受条件的随机(Random With Late Acceptance)等方法。

在纯随机超启发式算法中,每一个被选中的LLH即使不能改进问题实例的当前解,也总能应用于问题实例的求解。文献[3]将考试时间表问题(Exam Timetable Problem)转换为图着色问题(Graph Coloring Problem),然后利用图着色的两种LLH进行随机组合,并利用组合而成的启发式算法对实例进行求解。该超启发式算法与此前最好算法相比,所得解的质量有了一定的提高。

在蒙特卡罗超启发式算法中,若随机被选中的LLH不能改进当前问题实例解,则按照一定的概率(此概率通常根据“执行该LLH前后的”问题实例解的目标函数的值差计算)来决定是否采用该LLH。文献[4]运用蒙特卡罗超启发式算法就印刷电路板上的元器件部署问题(Component Placement Sequencing Problem)进行了求解。他们使用了6种LLH,当LLH不能改进当前实例解时,分别尝试了线性 and 指数概率函数来决定该LLH的取舍。实验结果表明,指数概率函数性能优于线性概率函数。此外,文献[5]也利用蒙特卡罗超启发式算法对训练者调度问题(Trainer Scheduling Problem)展开了研究,取得了比现有元启发式算法更好的效果。

文献[6]在频道分配问题(Channel Assignment Problem)上研究了带确定性接受条件的随机超启发式算法。每当一个LLH被选中时,若它能改进当前解,则被采用;若所得解超过一个指定的阈值,也可被采用。在超启发式算法执行的初始阶段,该阈值设定为初始解的目标函数值。随着超启发式算法的迭代执行,该阈值按照一个固定的比例下降。利用这种策略,在超启发式算法执行初期,尽管那些不能改进当前解的LLH容易被采纳,但随着执行进程不断深入,它们会越来越少的被采纳。

文献[7]研究了考试时间表问题中带延迟接受条件的超启发式算法。在该工作中,超启发式算法由一系列LLH随机组合形成新的启发式算法。每

一个被选中的LLH将用于当前解的优化,若所得到的新解质量比M(M是预先设定的一个参数)步前的解质量更好,则该LLH被采纳。实验结果表明,延迟接受条件可以有效地改善随机超启发式算法的性能。

## 基于贪心策略的超启发式算法

该类超启发式算法在构造新启发式算法时,每次都挑选那些能够最大化改进当前(问题实例)解的LLH。由于每次挑选LLH时需要评估所有LLH,故此该类方法的执行效率低于基于随机选择的超启发式算法。

文献[8]针对个人调度问题(Personal Scheduling Problem),提出了一种贪心超启发式算法。他们通过贪心策略,每次从所有的LLH中挑选性能最好的一个用于改进当前解。实验结果表明,这种方法虽然需要更长的时间,但能提供高质量的解。

此外,文献[9]针对训练者调度问题,提出了一种贪心超启发式算法的变种形式。在构造新启发式算法时,该变种每次从所有LLH中挑选最好的一批构成一个候选列表,然后从该候选LLH列表中随机选用一个。在文献[9]中,候选列表的长度可以动态调整,以保持多样性(Diversification)和强化性(Intensification)的平衡。

## 基于元启发式算法的超启发式算法

该类超启发式算法采用现有的元启发式算法(作为高层次启发式方法)来选择LLH。这些元启发式算法包括蚁群算法、禁忌算法、可变邻域搜索和遗传算法等。

文献[10]研究了旅行锦标赛问题(Traveling Tournament Problem)中基于蚁群的超启发式算法。他们采用蚁群算法来管理和操纵LLH以获得新的启发式算法,每只蚂蚁均构造一个新的启发式算法。在算法构造过程中,这些蚂蚁之间进行信息交互,以获得更好的新启发式算法。文献[11]研究了P-中位问题(P-Median)上的带空间归约的蚁群超启发式算法。文中将LLH分为强化型(Intensification)和多样

型 (Diversification) 两类, 每次调用时, 先调用一个强化型LLH, 再调用一个多样型LLH。实验结果表明, 该方法在实现搜索空间归约的同时, 依然能提供高质量的解。

文献[12]针对教育时间表问题 (Educational Timetable Problem) 提出了基于禁忌搜索的超启发式算法。文中采用禁忌搜索方法来获得新的启发式算法, 其中每个新启发式算法均是由5种LLH搭配组合而成。2009年, 文献[13]进一步研究了教育时间表问题, 分别实现了基于最深梯度下降方法 (Steepest Descent Method)、迭代局部搜索 (Iterated Local Search) 以及可变邻域搜索的超启发式算法, 并与基于禁忌搜索的超启发式算法进行了对比。

文献[14]针对考试时间表问题, 研究了基于遗传算法的超启发式算法。种群中的每一个体由5种LLH组合构成。他们发现这种超启发式算法比其他类超启发式算法更容易发现可行解。他们还研究了种群中个体的编码方式对于解质量的影响, 发现可变长度编码优于固定长度编码。文献[15~16]提出了一种基于遗传算法的超启发式算法, 用于求解二维条带装箱问题 (2-D Strip Packing Problem)。该超启发式算法基于4种LLH, 采用染色体来对这些LLH进行编码演化。他们将超启发式算法与现有专门算法进行了对比, 实验结果表明在很多测试实例上, 超启发式算法能发现高质量的解。

文献[17]研究了约束可满足性问题 (Satisfiability Problem, SAT) 的基于遗传编程的超启发式算法。他首先将目前解决SAT问题的著名局部搜索算法WalkSAT和Novelty分解为一些基本模块, 然后采用遗传编程技术对这些模块重新组合, 以获得新的启发式算法。实验结果表明, 超启发式算法能获得比Walksat、Novelty及传统演化算法方法质量更好的解。文献[18]设计了针对二维条状装箱问题的基于遗传编程的超启发式算法。该算法对现有的启发式算法进行分解, 利用分解后的模块进行演化以获得新的启发式算法。实验结果表明, 该超启发式算法具有良好的通用性, 无需针对不同实例进行复杂的参数设置, 即可获得高质量解。

## 基于学习的超启发式算法

该类超启发式算法在构造新启发式算法时, 采用某种学习机制, 根据现有各种LLH的历史信息决定采纳哪种LLH。根据LLH历史信息来源的不同, 该类超启发式算法可以进一步分为在线学习 (On-Line Learning) 和离线学习 (Off-Line Learning)。前者是指LLH的历史信息在求解实例过程中积累下来的; 后者通常将实例集合分为训练实例和待求解实例两部分, 训练实例主要用于积累LLH的历史信息, 待求解实例可以根据历史信息来决定LLH的取舍。

在基于在线学习的超启发式算法研究中, 文献[19]采用强化学习 (Reinforcement Learning) 来选择LLH。每一个LLH都被赋予一定的分值, 每次根据分值来决定选择哪一个LLH。一旦一个LLH被选中, 它将用于优化当前解。若它能改进当前解, 则调高其分值, 否则降低其分值。文献[20]采用一种带禁忌搜索的强化学习 (Reinforcement Learning With Tabu Search) 作为在线学习机制, 用于选择LLH。除了每一个LLH被赋予分值外, 超启发式算法还采用了一个动态禁忌列表。在构造新启发式算法过程中, 每次挑选分值最高且不在禁忌列表中的LLH。一旦该LLH被选中, 它将被用于优化当前解。若它能改进当前解, 其分值被调高, 否则分值将被降低, 且被列入禁忌表中。

在基于离线学习的超启发式算法研究中, 文献[21]研究了如何利用关联分类方法来指导超启发式算法过程。由于超启发式算法在构造解的每一步都需要决定采用哪一个LLH, 因此如何选择LLH将会明显地影响最后的求解质量。文中根据各种LLH在训练实例方面的性能, 构造出了关联分类器, 对于待求解实例, 每次需要选择LLH时, 该算法则利用当前解的状态作为输入, 利用关联分类器预测LLH以获得最佳效果。

## 超启发式算法研究展望

虽然超启发式算法研究已经取得了很多人振



奋的成果，但是围绕它的算法构造和实例求解阶段还存在一系列重要的问题亟待解决。

## 算法构造阶段：超启发式算法搜索空间的特征分析及应用问题

超启发式算法与启发式算法具有一定的相似性：它们都是在高维（搜索）空间上进行查找。所不同的是，超启发式算法的搜索空间上每个点代表LLH的组合，而启发式算法搜索空间上的每个点代表问题实例的解。因此，针对启发式算法的研究思路对于超启发式算法具有很好的参考意义。

在传统启发式算法的研究中，很多研究者针对优化组合问题的搜索空间进行特征分析，并利用这些特征设计了多种高效算法。这种方法被认为是应对组合爆炸的根本出路之一<sup>[22]</sup>。目前刻画搜索空间常用手段是适应度地貌（Fitness Landscape）分析。在适应度地貌中，最重要的特征之一是所谓的“大坑”结构（Big Valley），即众多优化组合问题的局部最优解与全局最优解差别很小。形象地比喻，是以全局最优解为坑底，周围遍布大量的局部最优解。“大坑”结构最早是波俄思（Boese）<sup>[23]</sup>就旅行商问题（Traveling Salesman Problem）观察到的。他发现很多旅行商问题实例的局部最优解与全局最优解存在80%左右的重合边。随后文献[24]在SAT问题以及文献[25]在资源受限的项目调度问题（Resource Constrained Project Scheduling Problem）方面也发现了类似的特征。因此，人们开始考虑把多个局部最优解的相同部分固定住，从而把实例归约成规模更小的实例来进行求解。在这种研究思路下，许多研究者提出了多种高效启发式算法。例如，文献[26]给出了SAT问题的动态局部搜索算法；文献[27]给出了旅行商问题的归约算法；文献[28~29]给出了二次分配问题（Quadratic Assignment Problem）和图的划分问题（Graph Partitioning Problem）的基于归约的高效启发式算法。

与传统启发式算法相比，超启发式算法搜索空间的特征分析及应用研究工作还十分匮乏。超启发式算法研究还可以分解为两个相关科学问题。

**如何分析超启发式算法的搜索空间特征** 目前关于超启发式算法搜索空间的适应度地貌分析的成果很少，仅文献[30~31]在国际上首次进行过这方面的探索，文章以生产调度问题（Production Scheduling Problem）和时间表问题为例，通过实验发现超启发式算法的搜索空间也存在类似“大坑”结构，即最优的启发式算法附近聚集了大量“局部最优”的算法<sup>3</sup>。对于更多经典的优化组合问题，应该如何定义超启发式算法的适应度地貌、如何来分析其特征依然有待深入地研究。

**如何利用搜索空间特征来设计新超启发式算法** 在传统启发式算法的研究中，很多学者针对搜索空间特征给出了新的高效算法<sup>[26~29]</sup>。由于超启发式算法搜索空间也存在类似特征（比如“大坑”结构），如何利用这种结构特征来设计超启发式算法（主要是指其中的高层次启发式方法）是有趣而重要的课题，目前国际上还没有这方面的成果报道。由于超启发式算法的搜索空间中每个点代表一组LLH的组合，因此多个“局部最优”的启发式算法的共同部分，实际是若干个LLH的固定搭配模式。参照传统启发式算法的研究方法，利用固定搭配模式对超启发式算法的搜索空间进行收缩，可以更有效地促进启发式算法生成。但具体如何收缩超启发式算法的搜索空间，还有待进一步研究。

## 实例求解阶段：基于实例特征的算法选择问题

现实应用中通常需要反复求解同一个问题的大量实例，这对于传统启发式算法和超启发式算法具有完全不同的影响。例如，在从事超大规模集成电路（Very Large Scale Integration, VLSI）设计的公司中，用户需求的变化和产业的升级会导致同一个

<sup>3</sup> 与传统启发式算法的搜索空间不同，超启发式算法搜索空间上的每一个点对应一种启发式算法，即一组LLH的组合。

问题（如超大规模集成电路的布线问题）的实例不断改变，故此需要不断求解新实例。这种应用模式对于传统启发式算法而言，只是已有算法的重复使用，而对于超启发式算法却具有显著不同的意义。其原因在于：超启发式算法会针对每一个实例来构造算法，从而导致：（1）面对一组不同实例，超启发式算法会产生大量的新启发式算法；（2）求解单个实例的开销较大，因为必须由LLH逐个组合得到合适的新启发式算法。因此，如何利用超启发式算法在求解同一问题实例上积累的历史信息，对于该领域的研究具有重要意义。

目前，超启发式算法对历史信息的利用还很不充分。基于随机、基于贪心和基于元启发式算法的三类超启发式算法均未考虑到LLH历史信息的使用。基于学习的超启发式算法（包括在线学习和离线学习）也仅仅部分运用了历史信息，仍然存在一些明显不足。在线学习的超启发式算法<sup>[19~20]</sup>仅仅考虑当前实例的求解，而不存储和利用以往的实例求解信息，它所学习的是各个LLH在求解当前实例的不同阶段中表现出来的性能<sup>4</sup>。尽管离线学习的超启发式算法<sup>[21]</sup>利用了以往的实例求解信息来引导当前实例的求解，但现有的求解模式还存在一定的缺陷。首先，在待求解实例上，离线学习的超启发式算法依然由一系列迭代构成，每次迭代利用当前解的状态作为输入，利用关联分类器来预测一个LLH，以获得最佳效果。由于一个完整的启发式算法需要由很多LLH组合而成，也就意味着需要反复调用关联分类器才能得到一个完整的启发式算法，因此离线学习的超启发式算法时间开销较大；其次，这种做法类似于一系列局部优化，并不能保证最终获得高质量的新启发式算法（每一步获得好的LLH不能确保最终的启发式算法是高效的）。

根据上述分析，若要充分利用历史信息来高效求解实例，应该以完整的启发式算法作为基本单元

来进行学习和预测。这种新模式同样需要将实例分为训练实例和待求解实例两部分：在训练实例上积累历史信息，并利用该信息引导求解新实例。新模式将每次在训练实例上获得的新启发式算法作为一个整体来考虑，构造学习器和运用学习器时均采用完整启发式算法作为基本单元。这种新模式与离线学习的超启发式算法存在本质差别。与现有离线学习的超启发式算法相比，这种模式具有以下优势：

- （1）利用新实例作为输入，只需运用一次学习器即可得到完整启发式算法，从而节约算法产生时间；
- （2）由于是以完整启发式算法为基本单元，可以在一定程度上避免算法构造的“局部最优”情况。■



江贺

CCF 会员。大连理工大学副教授。主要研究方向为智能计算及其应用。  
jianghe@dlut.edu.cn

## 参考文献

- [1] Wolpert D.H., Macready W.G. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*. 1997, 1(1): 67 ~ 82
- [2] Burke E.K., Hyde M., Kendall G., Ochoa G., Ozcan E., Woodward J. A classification of hyper-heuristics approaches. *Handbook of Metaheuristics*, International Series in Operations Research & Management Science, In M. Gendreau and J-Y Potvin (Eds.), Springer (in press). 2009
- [3] Qu R., Burke E.K., McCollum B. Adaptive automated construction of hybrid heuristics for exam timetabling and graph colouring problems. *European Journal of Operational Research (EJOR)*, 2009, 198(2): 392 ~ 404
- [4] Ayob M., Kendall G. A Monte Carlo hyper-heuristic to optimise component placement sequencing for multi head placement machine. In: *Proceedings of the International Conference on Intelligent Technologies (InTech'03)*, 2003, 132 ~ 141

<sup>4</sup> 这种模式下，通常是根据当前实例和给定的初始解进行若干次迭代优化，因此每次迭代中各个LLH会有不同的性能，而这些信息可以用来引导随后的迭代过程。对于第 $i$ 次迭代过程而言，过去的1至 $i-1$ 次迭代积累下来各个LLH的性能数据，则通过一定的学习手段（如强化学习）可以决定哪一个LLH有希望产生最好的性能。

- [5] Chakhlevitch K. A hyperheuristic methodology for real-world scheduling. PhD Thesis, Department of Computing, University of Bradford, UK, 2006
- [6] Kendall G., Mohamad M. Channel assignment in cellular communication using a great deluge hyper-heuristic. In: Proceedings of IEEE International Conference on Network (ICON2004), 2004, 769 ~ 773
- [7] Ozcan E., Bykov Y., Birben M., Burke E.K. Examination timetabling using late acceptance hyper-heuristics. In: Proceedings of IEEE Congress on Evolutionary Computation (CEC 2009), 2009, 997 ~ 1004
- [8] Cowling P., Kendall G., Soubeiga E. Hyperheuristics: a tool for rapid prototyping in scheduling and optimisation. In Applications of Evolutionary Computing: Proceeding of Evo Workshops 2002 (Eds. Cagoni, S., Gottlieb, J., Hart, E., Middendorf, M. & Goenther, R.), 2002, 1 ~ 10
- [9] Cowling P., Chakhlevitch K. Using a Large set of low level heuristics in a hyperheuristic approach to personnel scheduling. Evolutionary Scheduling, Studies in Computational Intelligence, (Eds. K. P. Dahal, K. C. Tan, P.I. Cowling), Heidelberg: Springer-Verlag, 2007, (49): 543 ~ 576
- [10] Chen P.C., Kendall G., Vanden-Berghe G. An ant based hyper-heuristic for the travelling tournament problem. In: Proceedings of IEEE Symposium of Computational Intelligence in Scheduling (CISched 2007), Hawaii, 2007, 19 ~ 26
- [11] Ren Z, Jiang H., Xuan J., Luo Z. Ant Based Hyper Heuristics with Space Reduction: A Case Study of the p-Median Problem. Proceedings of 11th International Conference on Parallel Problem Solving From Nature (PPSN 2010), Krakow, Poland. September 11-15, 2010, pp. 546 ~ 555
- [12] Burke E.K., McCollum B., Meisels A., Petrovic S., Qu R. A graph-based hyper heuristic for educational timetabling problems. European Journal of Operational Research, 2007, 176(1): 177 ~ 192
- [13] Qu R., Burke E.K. Hybridisations within a graph based hyper-heuristic framework for university timetabling problems. Journal of the Operational Research Society, 2009, 60: 1273 ~ 1285
- [14] Pillay N. An analysis of representations for hyper-heuristics for the uncapacitated examination timetabling problem in a genetic programming system. In: Proceedings of the 2008 Annual Conference of the South African Institute of Computer Scientists and Information Technologists on IT Research in Developing Countries, SAICSIT Conf. 2008, ACM International Conference Proceeding Series 338, New York: ACM, 2008, 188 ~ 192
- [15] Garrido P., Riff M.C. An evolutionary hyperheuristic to solve strip-packing problems. In: Proceedings of Intelligent Data Engineering and Automated Learning - IDEAL 2007, 8th International Conference, Lecture Notes in Computer Science 4881, Heidelberg: Springer-Verlag, 2007, 406 ~ 415
- [16] Garrido P., Riff M.C. Collaboration between hyperheuristics to solve strip-packing problems. In: Proceedings of Foundations of Fuzzy Logic and Soft Computing, 12th International Fuzzy Systems Association World Congress, IFSA 2007, Lecture Notes in Computer Science 4529, Heidelberg: Springer-Verlag, 2007, 698 ~ 707
- [17] Fukunaga A.S. Automated discovery of local search heuristics for satisfiability testing. Evolutionary Computation, 2008, 16(1): 21 ~ 61
- [18] Burke E.K., Hyde M., Kendall G., Woodward J. A genetic programming hyper-heuristic approach for evolving two dimensional strip packing heuristics. IEEE Transactions on Evolutionary Computation (to appear), 2010
- [19] Nareyek A., Smith S.F., Ohler C.M. Integrating local-search advice into refinement search (or not). In: Proceedings of the CP 2003 Third International Workshop on Cooperative Solvers in Constraint Programming, 2003, 29 ~ 43
- [20] Burke E.K., Kendall G., Soubeiga E. A tabu-search hyperheuristic for timetabling and rostering. Journal of Heuristics, 2003, 9(6): 451 ~ 470
- [21] Thabtah F, Cowling P. Mining the data from a hyperheuristic approach using associative classification. Expert Systems with Applications: An International Journal, 2008, 34(2): 1093 ~ 1101
- [22] 李国杰. 对计算机科学的反思. 2005年中国计算机大会
- [23] Boese K.D. Cost versus distance in the traveling salesman problem. Technical Report, CSD-950018, Los Angeles: UCLA, 1995
- [24] Zhang W. X. Configuration landscape analysis and backbone guided local search: Part I: satisfiability and maximum satisfiability. Artificial Intelligence, 2004, 158(1): 1 ~ 26
- [25] Jens C., Andreas F. Fitness landscape analysis for the resource constrained project scheduling problem. In: Proceedings of 2009 International Conference on Learning and Intelligent Optimization, London: Springer, 2009, 104 ~ 118
- [26] Valnir F. Backbone guided dynamic local search for

- propositional satisfiability. In: Proceedings of the 9th International Symposium on Artificial Intelligence and Mathematics (AI & Math -06), New York: Springer, 2006, 100 ~ 108
- [27] Richter D., Goldengorin B., Jäger G., Molitor P. Improving the efficiency of Helsgaun's Lin-Kernighan heuristic for the symmetric TSP. In: J. Janssen and P. Pralat, eds. Proc. of the CAAN'07, Lecture Notes in Computer Science 4852, Heidelberg: Springer-Verlag, 2007. 99 ~ 111
- [28] 江贺, 张宪超, 陈国良, 李明楚. 二次分配问题的骨架分析与算法设计. 中国科学 (E辑), 2008, 38(2): 209 ~ 222
- [29] 江贺, 邱铁. GPP问题的骨架分析与启发式算法设计. 计算机学报, 2009, 32(8): 1662 ~ 1667
- [30] Ochoa G., Qu R., Burke E.K. Analyzing the landscape of a graph based hyper-heuristic for timetabling problems. In Proceedings of Genetic and Evolutionary Computation Conference (GECCO-09), New York: ACM, 2009, 341 ~ 348
- [31] Ochoa G., Vazquez-Rodriguez J.A., Petrovic S., Burke E.K. Dispatching rules for production scheduling: a hyper-heuristic landscape analysis. In Proceedings of the IEEE Congress on Evolutionary Computation (CEC-09), IEEE Press, Trondheim, Norway. 2009, 1873 ~ 1880