

基于改进协同遗传算法的有效载荷系统功能序列规划方法*

王 静^{1,2}, 王春梅¹, 姚秀娟¹

(1. 中国科学院国家空间科学中心, 北京 100190; 2. 中国科学院大学, 北京 100049)

摘要:针对传统回溯算法在求解基于知识模型的有效载荷系统功能序列规划问题中搜索效率低的问题,提出一种基于“择劣变异”(Worst Individual Mutation, WIM)策略的协同遗传算法(Co-evolutionary Genetic Algorithm, CGA)的改进算法 WIM-CGA。该算法在遗传过程中采用双路线进化方案,即“择优实施标准遗传过程,择劣实施变异操作”,达到提高求解精确度及搜索效率的目的。仿真结果表明,同等测试条件下,当功能规模为50,约束密度为1.0时,WIM-CGA算法在限定时间内最优解的平均精确度比优化的回溯算法提高了54.15%,比CGA算法提高了6.18%,且当所得解的精确度大于90%时,WIM-CGA算法比CGA算法的迭代次数减少了65.79%,耗时降低了48.97%,显著提高了功能序列规划的效率。

关键词:知识模型;功能序列规划;协同遗传算法;择劣变异;生存期适应度评估

中图分类号:TP391 **文献标志码:**A **文章编号:**1001-2486(2019)06-019-06

Functional sequence planning method based on improved co-evolutionary genetic algorithm for payload system

WANG Jing^{1,2}, WANG Chunmei¹, YAO Xiujuan¹

(1. National Space Science Center, Chinese Academy of Sciences, Beijing 100190, China;

2. University of Chinese Academy of Sciences, Beijing 100049, China)

Abstract: Aiming at the low search efficiency of the traditional backtracking algorithm when planning the function sequence of the payload system based on the knowledge model, an improved algorithm named as WIM-CGA for CGA (co-evolutionary genetic algorithm) was proposed, which was based on the WIM (worst individual mutation) strategy. The algorithm adopted a dual-route evolution scheme in the genetic process, which was “the better individuals perform standard genetic processes, and the worse individuals perform mutation operation”, to improve the solution accuracy and search efficiency. Simulation results show that under the same test conditions, when the function scale is 50 and the constraint density is 1.0, the average accuracy of the optimal solution of WIM-CGA within the limited time is 54.15% higher than that of GAC-BS (BS based on generalized arc consistency) and 6.18% higher than CGA, and when optimal solution accuracy reaches 90%, the iteration times of WIM-CGA is 65.79% lower than that of CGA, and the time consumed is reduced by 48.97%. The efficiency of functional sequence planning is improved significantly.

Keywords: knowledge model; functional sequence planning; co-evolutionary genetic algorithm; worst individual mutation; life time fitness evaluation

有效载荷系统测试用例自动规划是航天测试领域的重要课题^[1-2]。在地面综合测试中,有效载荷系统某一功能的测试依赖于—组满足特定执行逻辑约束的活动(封装了指令、影响等)序列,有效载荷系统知识模型中将功能定义为活动与约束两部分,而要形成有效的测试序列,一种基于约束知识的功能序列规划方法必不可少。

依据约束规划功能序列问题可转换为约束满足问题(Constraint Satisfaction Problem, CSP)^[3-5]。

目前,针对CSP,基于回溯的搜索算法(Backtracking Search, BS)是完备性求解方法^[5],但当功能规模大于10,约束密度大于0.5时,其搜索效率极低,难以在合理时间内求得最优解,基于广泛弧相容(Generalized Arc Consistency, GAC)技术压缩搜索空间未有明显改善。搜索过程中反复的约束检查是求解缓慢的重要原因,为减少约束检查频次,Paradis提出基于生存期适应度评估(Life Time Fitness Evaluation, LTFE)的协同进化遗传算法

* 收稿日期:2018-07-18

基金项目:国防科技工业民用专项科研工程研制资助项目(Y76602FH6S)

作者简介:王静(1990—),女,山东威海人,博士研究生,E-mail:wangj711@yeah.net;

王春梅(通信作者),女,研究员,学士,博士生导师,E-mail:wangchm@nssc.ac.cn

(Co-evolutionary Genetic Algorithm, CGA)^[6-7], 将个体对部分选中约束的满足度作为个体适应度的有效评估。尽管 CGA 算法能够有效降低计算复杂度, 但容易早熟收敛, 难以求得最优解或精确度较高的近似解, 因此需进一步提升 CGA 算法的搜索能力, 使其在较短时间内求得最优解(或高精度的近似解)。

1 功能知识模型描述

功能知识模型是有效载荷系统知识模型(Knowledge Model of Payload System, KMPS)的一个重要部分。KMPS 基于测试用例自动规划的需求构建, 将知识空间划分为五个维度, 全面描述其领域知识: ①系统级知识, 抽象描述系统顶层知识, 将系统拆分成多个系统元素; ②元素级知识, 概述系统元素的组成成分; ③行为级知识, 详细描述系统元素的行为特征; ④元级知识, 描述其他维度中未细化的底层知识; ⑤推理级知识, 描述规划测试用例的逻辑性知识, 与测试需求相关。KMPS 基于本体^[8-10]建模, 非本文研究重点, 不做详细说明。

功能知识属于行为级知识, 功能设计的特点在于, 以活动(封装指令、影响等)为驱动, 并且活动间需满足特定的执行逻辑要求等约束, 因此将其知识模型定义为三元组:

$$function = (P, A, C)$$

- *P*: 功能基本属性, 如名称、ID;
- *A*: 功能中的活动, 封装了指令、该指令执行的逻辑约束以及执行后的影响等;
- *C*: 实现功能所需满足的各类约束, 如功能逻辑约束、指令参数取值约束等。

功能序列规划问题, 即求解满足 *A* 和 *C* 中“逻辑约束”的序列, 表 1 展示某星上有效载荷系统两类约束的示例(有“”的表示活动):

1) 功能逻辑约束, 描述功能内所特有的约束, 定义为三元组 $FLC = (T, R, E)$, 其中 *T* 为靶活动, 即被约束的活动; *R* 为约束条件类型, 包含前置、后置、直接前置、直接后置四种; *E* 为逻辑表达式, 如: $E = A_i \wedge A_j$ (A_i, A_j 为活动变量)。令 *O* 表示 *E* 中所含活动的集合, 则 *O* 与 *T* 须满足 $O \cap T = \emptyset$ 。

2) 活动逻辑约束, 表示活动在所有功能中均需满足的固有约束, 封装在活动中, 其所属的活动本身即为“靶活动”, 因而定义为二元组: $ALC = (R, E)$, 变量含义与功能逻辑约束类似。

表 1 某星上有效载荷系统的两类约束示例

Tab. 1 Examples of two kinds of constraints on payload system of a satellite

逻辑约束类型	约束示例
功能逻辑约束	“相机拍照”前须先执行“设置通道”和“设置曝光”
活动逻辑约束	“设备 C 开机”前须先执行“设备 A 开机”或“设备 B 开机”

基于上述功能知识模型, 可将功能序列规划问题描述如下:

设某功能中包含 x_1, \dots, x_n 共 *n* 个活动, 其中 *m* 个活动具有逻辑约束 C_1, C_2, \dots, C_m , 设活动 x_i 在功能序列中的位置为 D_i , 则功能序列规划的问题模型可描述为三元组:

$$I = (X, D, C)$$

其中, $X = \{x_1, x_2, \dots, x_n\}$ 为 *n* 个变量的集合, 即 *n* 个活动; $D = \{D_1, D_2, \dots, D_n\}$ 为 *n* 个变量取值的集合, 即活动在功能中的位置值, $D_i \in \{1, 2, \dots, n\}$; $C = \{C_1, C_2, \dots, C_m\}$ 为约束集合, $C_i = (X_i, R_i)$, C_i, X_i 及 R_i 之间的对应关系示例如表 2 所示, 其中, $X_i = \{x_{i_1}, \dots, x_{i_{k_i}}\}$ 是变量 *X* 的子集, 长度为 k_i , R_i 为约束关系, 是 $D_{i_1} \times \dots \times D_{i_{k_i}}$ 的子集。

若存在映射 $f: X \rightarrow D$, 对 $\forall (X_i, R_i), i = 1, 2, \dots, m$, 均满足:

$$f(X_i) = \{f(x_{i_1}), f(x_{i_2}), \dots, f(x_{i_{k_i}})\} \in R_i$$

则 $f(X) = \{f(x_1), f(x_2), \dots, f(x_n)\}$ 为问题 *I* 的一个解。

上述模型符合约束满足问题^[11-12]的数学描述, 由此, 基于知识模型的功能序列规划问题转换为约束满足问题, 结合问题实际情况, 定义约束密度及约束松紧度的概念如下:

定义 1 约束密度 (Constraint Density, CD), 一个功能的约束密度是指, 该功能中绑定约束的活动数占功能中活动总数的比例。

定义 2 约束松紧度 (Constraint Tightness, CT), 指约束表达式 (*E*) 中涉及的活动个数。

表 2 C_i 与 X_i 和 R_i 的对应关系示例

Tab. 2 Examples of C_i corresponding to X_i and R_i

C_i	X_i	R_i
x_3 的前置 为 x_2 和 x_8	(x_3, x_2, x_8)	$\{(D_2, D_3, D_8) \mid (D_3 > D_2) \wedge (D_3 > D_8)\}$
x_9 的后置 为 x_3 或 x_5	(x_9, x_3, x_5)	$\{(D_3, D_5, D_9) \mid (D_9 < D_3) \vee (D_9 < D_5)\}$

2 GAC-BS 与 CGA 算法的搜索性能

对于 CSP 问题,回溯搜索 (Backtracking Search, BS) 算法是一种完备的求解方法,理论上,若问题有解,则一定可以求出解,若问题无解,则可以证明无解^[13]。然而,面对复杂的实际问题,其搜索代价很高,通常难以在合理的时间内求解,为降低搜索空间,通常采用相容性技术^[14-15]进行预处理。针对多元约束的功能序列规划问题,应用基于广泛弧相容^[16-17]的回溯搜索算法 GAC-BS 进行处理,其求解效率仍然很低,在 CD 取 0.5, CT 取 1~3 的情况下,功能规模大于 10 后,搜索效率急速下降,当达到 20 时,1000 之内不能求得满足约束的可行解,且其求得的最优解平均仅能满足 46.59% 的约束。换言之,当问题规模较大时,通过压缩搜索空间的优化方式效果并不理想。

功能序列规划过程中反复的约束检查是求解缓慢的另一个重要原因,合理地减少约束检查次数可以有效提高搜索效率。针对这一思路,Paradis 提出基于生存期适应度评估 (Life Time Fitness Evaluation, LTFE) 的协同进化遗传算法 (Co-evolutionary Genetic Algorithm, CGA)^[18-19],其一次迭代过程如图 1 所示。

CGA 算法的优势在于:①“协同思想”将可能解与约束均看作个体,在进化过程中, sol 种群与 con 种群的适应度不断变化,一旦可能解满足约

束,则约束将获得更低的适应度,而尚未满足的约束则获得较高的适应度,具有更大的选中概率,最终使得可能解种群中的个体专注于满足“更难”的约束;②“LTFE 思想”使得可能解的适应度取自于对部分约束的满足情况的评估。由于约束的选择依据来自“协同思想”中约束种群的适应度值,所选约束具有代表性,能够有效评估适应度,从而使得可能解不需要计算对所有约束的满足情况,降低计算复杂度。

然而,CGA 算法在搜索过程中容易过早收敛,对于 CD 为 1.0, CT 为 1~3, 功能规模为 50 的情况下,要达到 90% 以上精确度,平均需迭代 2.29×10^4 次,且 500 s 内的迭代最优解仅能达到 90.79% 的精确度,搜索能力需进一步提升。

3 WIM-CGA 算法改进策略

针对 CGA 算法在迭代过程中过早收敛的问题,基于“择劣变异” (Worst Individual Mutation, WIM) 策略的思想,对 CGA 算法进行改进,称之为 WIM-CGA 算法,其改进思路为:在 sol 种群的遗传过程中新增一种“择劣变异”的进化方式,按照适应度低的个体优先选择的原则,从种群中选择两个适应度较低的个体,直接执行变异操作,并将变异后的新个体与当前种群中适应度最低的个体进行比较,保留两者中适应度较高的个体。因此, WIM-CGA 算法在遗传过程中采用了双路线进化方案,“择劣实施标准遗传过程,择劣实施变异操

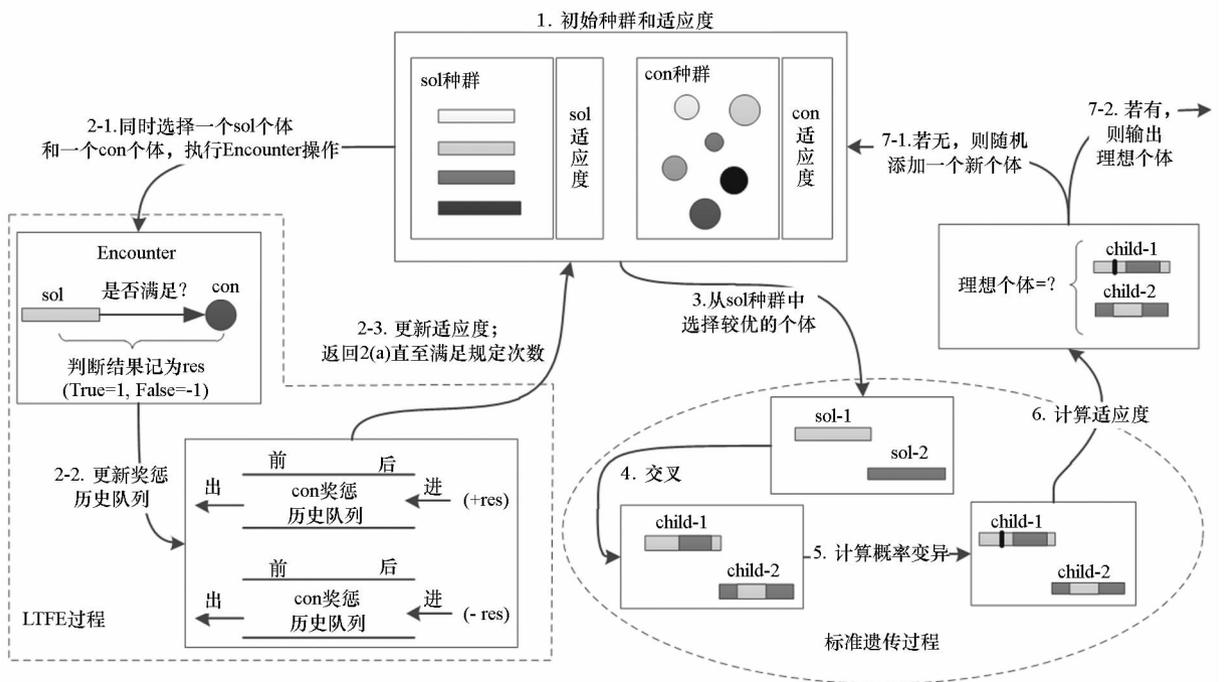


图1 CGA 算法一次迭代的流程

Fig. 1 Steps of one iteration of CGA

作”，在保留原 CGA 算法优势的基础上，变异较差的个体，增加基因的多样性，提高算法的局部搜索能力。WIM-CGA 算法的一次迭代流程如图 2 所示，将图 1 中第 3 步改为双路线，路线 3-1 执行图 1 中的标准遗传过程，路线 3-2 执行“择劣变异”过程。需要说明的是：①路线 3-2 不删除被选中个体；②步骤 5-1 中，“添加”操作采用两种方式，对于路线 3-1 产生的后代，“添加”方式不变(同图 1)，对于路线 3-2 产生的新个体，“添加”操作须将新个体与种群中适应度最差的个体相比，若优于最差个体，则添加新个体并删除最差个体，否则，删除新个体。

从 WIM-CGA 算法的迭代步骤中可以看出，理论上，改进算法 WIM-CGA 相比 CGA 算法，可提高种群的整体适应度。假设一次迭代中，被选中进行直接变异的个体，经变异操作后基因变差(相比种群中的最差个体)的概率为 p ，则变优的概率为 $1 - p$ 。由于“添加”操作不会将变异产生的差个体添加到种群中，因此种群适应度不受影响的概率为 p ，种群适应度变优的概率为 $1 - p$ 。由此， n 次迭代后，种群适应度不受影响的概率为 p^n ，变优的概率为 $1 - p^n$ ，当 $n \rightarrow \infty$ 时， $\lim_{n \rightarrow \infty} (1 - p^n) = 1$ 。

4 仿真与分析

仿真测试主要从两方面验证 WIM-CGA 算法的性能：一是求解效率，通过对比限定时间内的求

解成功率来反映；二是最优解的精确度，通过对比限定时间内所求最优解的约束满足度来体现。

仿真实验设计为：依次对功能规模(即活动个数)为 10,20,30,40,50 的功能测试序列进行测试，取 CT 为 1~3，且 CT 取 1,2,3 的比例依次为 25%，50%，25%。每一种功能规模生成 100 组指定 CD 的测试数据(即约束条件)，测试结果取其平均值，限制搜索时长为 300。表 3 中列出功能规模为 20，约束密度为 1.0 的第 7 组测试数据示例，其中参考序列的作用在于为随机生成约束条件提供参考，以保证所生成的约束问题有解；约束中，至少包含“pre”和“post”中的一个，表明约束类型，“expression”为约束的逻辑表达式，“operators”及“code”是生成表达式的辅助参数，分别表示“expression”中逻辑符号与活动编码，“target”表示靶活动。

表 3 仿真测试数据示例

Tab.3 Example of test data for simulation

项目	详细
编号	20/1.0/7(功能规模/密度/组数)
参考序列	["1", "4", "19", "13", "15", "16", "14", "6", "5", "10", "18", "11", "8", "7", "12", "0", "9", "3", "2", "17"]
约束列表	Constraint1 : { "post" : { "operators" : ["1"], "codes" : ["4", "9"] "expression" : "419" } "target" : "7" }

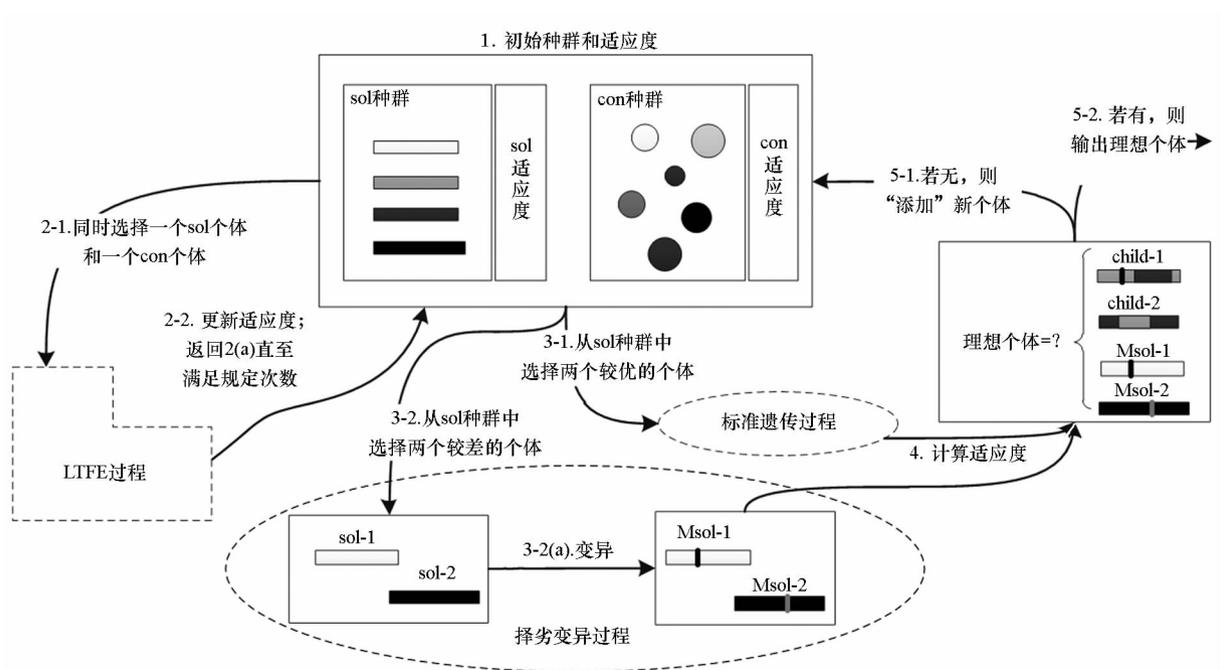


图 2 WIM-CGA 算法一次迭代流程

Fig.2 Steps of one iteration of WIM-CGA

1)求解成功率比较。分别对比 CD 为 0.5 和 CD 为 1.0 时,三种算法在限定时间内的求解成功率如图 3~4 所示。

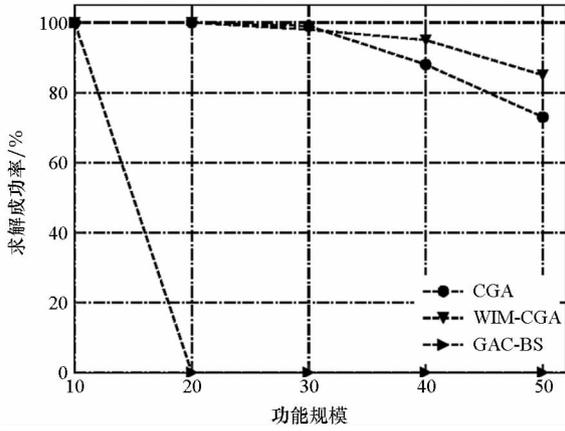


图 3 CD 为 0.5 时三种算法的求解成功率对比

Fig. 3 Comparison of the success rate of three algorithm when CD is 0.5

由图 3 可见,当 CD 为 0.5 时,三种算法在规模为 10 的情况下均可在时限内求解,而当规模增加后,GAC-BS 算法在时限内的求解成功率急速下降,当功能规模达到 20 时,求解成功率为 0。而 WIM-CGA 算法在功能规模达到 50 时,仍保证 85% 以上的求解成功率,明显优于 GAC-BS 算法,稍优于 CGA 算法。图 4 中,当 CD 增加到 1.0 时,三种算法相比于图 3 均有下降,但 WIM-CGA 算法仍优于另两种算法。

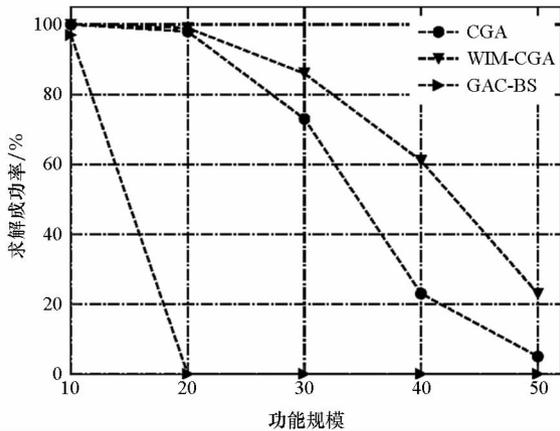


图 4 CD 为 1.0 时的求解成功率对比

Fig. 4 Comparison of the success rate of three algorithm when CD is 1.0

2)约束满足度比较。当功能规模与 CD 取值较大时,求解成功率不能全面评估各算法最终解的好坏,由此,计算三种算法在时限内搜索最优解的约束满足度 (Constraint Satisfaction Degree, CSD),以 CD 为 1.0 为例,对比结果如图 5 所示,

WIM-CGA 算法最优解的 CSD 为 96.74%,比 GAC-BS 高了 54.15%,比 CGA 算法高了 6.18%,进一步提升了 CGA 算法求解的精确度。

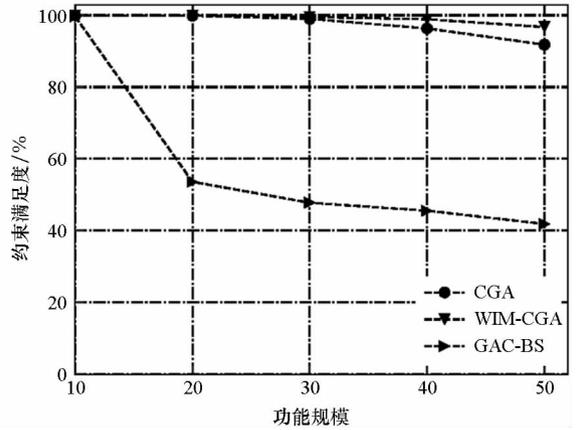


图 5 CD 为 1.0 时最优解的约束满足度对比

Fig. 5 Comparison of the constraint satisfaction rate of optimum solution when CD is 1.0

尽管 WIM-CGA 算法与 CGA 算法的最终解的 CSD 仅相差 6.18%,但 CGA 算法的搜索效率远低于 WIM-CGA 算法。对比 CGA 算法与 WIM-CGA 算法在问题规模为 50,CD 为 1.0 时,迭代过程中不同 CSD 对应的迭代次数及搜索耗时情况如表 4 和表 5 所示。

表 4 各 CSD 对应的迭代次数对比

Tab. 4 Comparison of the number of iterations corresponding to each CSD

算法	>70%	>80%	>90%	>96%
CGA	434	4585	22 916	
WIM-CGA	211	863	7839	12 900

表 5 各 CSD 对应的搜索时间对比

Tab. 5 Comparison of the search times corresponding to each CSD

算法	>70%	>80%	>90%	>96%
CGA	4	49	243	
WIM-CGA	2	11	124	198

由表 4 和表 5 中数据可见,在达到同样级别的 CSD 时,WIM-CGA 算法所需迭代次数及搜索时间均明显低于 CGA 算法,当 CSD 大于 90% 时,WIM-CGA 算法的迭代次数比 CGA 算法减少了 65.79%,耗时降低了 48.97%,并且 WIM-CGA 算法在时限内均能保证 96% 以上的 CSD。

综上,所提改进算法 WIM-CGA 的搜索效率、

最优解的精确度均优于 CGA 算法和 GAC-BS 算法,且能够有效改善 CGA 算法过早收敛、搜索缓慢的不足,提高求解速率及近似解的精确度。

目前实际应用中,功能规模通常不超过 25, CD 小于 0.5,在 CT 取 1~3 的情况下,WIM-CGA 算法最多耗时 7.8 s 即可求得最优解,而对于更为复杂的情况,根据仿真实验结果可知,WIM-CGA 算法平均 200 s 内可提供 96% 以上精确度的近似解。

5 结论

研究基于有效载荷系统知识模型的功能序列规划方法,设计“择劣变异”策略改进基于 LTFE 的 CGA 算法的 WIM-CGA 算法,改善回溯搜索算法在功能规模较大、约束较为复杂的情况下搜索效率极低的问题。该算法在 CGA 算法的基础上新增一种进化方式,采用“择优实施标准遗传过程,择劣实施变异操作”的双路线进化方案,在保留 CGA 算法原有优势的前提下,通过直接变异较差的个体,增加种群基因的多样性,提升种群的整体质量,改善 CGA 算法易于早熟收敛的不足,提升 CGA 算法的局部搜索能力。仿真结果表明,WIM-CGA 算法相比 GAC-BS 算法和 CGA 算法,显著提高搜索效率,具有明显优势。

参考文献 (References)

[1] 贺宇峰,赵光恒,郭丽丽,等. 基于改进 OOPN 的有效载荷系统测试用例生成方法[J]. 系统工程与电子技术[J]. 2010, 32(11): 2470-2475.
HE Yufeng, ZHAO Guangheng, GUO Lili, et al. Automatic generation of test cases for payload system based on improved OOPN [J]. Systems Engineering and Electronics, 2010, 32(11): 2470-2475. (in Chinese)

[2] 吕笑慰,王华茂,闫金栋. 基于状态图的航天器测试用例设计[J]. 航天器工程, 2014, 23(6): 135-140.
LYU Xiaowei, WANG Huamao, YAN Jindong. Research of state charts-based test case design in spacecraft test [J]. Spacecraft Engineering, 2014, 23(6): 135-140. (in Chinese)

[3] Bulatov A A. Constraint satisfaction problems; complexity and algorithms [C]//Proceedings of International Conference on Language and Automata Theory and Applications, Berlin: Springer, 2018: 1-25.

[4] Jiang X, Zhao Y T, Xu R, et al. An action-based constraint satisfaction algorithm for planning problems [C]//Proceedings of International Conference on Mechatronics and Intelligent Robotics, Berlin: Springer, 2017: 412-417.

[5] 张永刚,张思博,薛秋实. 求解约束满足问题的改进蚁群优化算法[J]. 通信学报, 2015, 36(5): 2015123-1-2015123-7.
ZHANG Yonggang, ZHANG Sibao, XUE Qiushi. Improved ant colony optimization algorithm for solving constraint

satisfaction problem [J]. Journal on Communications, 2015, 36(5): 2015123-1-2015123-7. (in Chinese)

[6] Paredis J. Exploring the evolution of genotype phenotype mappings [C]//Proceedings of IEEE Congress on Evolutionary Computation(CEC), 2017: 279-285.

[7] Paredis J. Co-evolutionary constraint satisfaction [C]//Proceedings of International Conference on Parallel Problem Solving from Nature, Heidelberg: Springer, 1994, 886: 46-55.

[8] Manaa M, Akaichin J. Ontology-based modeling and querying of trajectory data [J]. Data & Knowledge Engineering, 2017(111): 58-72.

[9] Alirezaie M, Kiselev A, Långkvist M, et al. An ontology-based reasoning framework for querying satellite images for disaster monitoring[J]. Sensors, 2017, 17(11): 2545-2569.

[10] 邓志宏,老松杨,白亮,等. 一种可扩展的 Cyber 空间作战体系描述模型[J]. 国防科技大学学报, 2014, 36(1): 184-190.
DENG Zhihong, LAO Songyang, BAI Liang, et al. An extensible description model of cyber war system[J]. Journal of National University of Defense Technology, 2014, 36(1): 184-190. (in Chinese)

[11] Hao X X, Liu J. A multiagent evolutionary algorithm with direct and indirect combined representation for constraint satisfaction problems [J]. Soft Computing, 2017, 21(3): 781-793.

[12] Dapic P, Markovic P, Martin B. Quantified constraint satisfaction problem on semicomplete digraphs [EB/OL]. New York: Association for Computing Machinery [2018-01-24]. <http://dro.dur.ac.uk/20046/1/20046.pdf>.

[13] Li B B, Li D, Raijai M, et al. Backtracking algorithm-based disassembly sequence planning [EB/OL]. Amsterdam: Elsevier[2018-03-12]. <http://www.sciencedirect.com/science/article/pii/S2212827117309393>.

[14] 张永刚,程竹元. 最大受限路径相容约束传播算法的研究进展[J]. 计算机科学, 2018; 45(6A): 41-45, 62.
ZHANG Yonggang, CHENG Zhuyuan. Research progress on max restricted path consistency constraint propagation algorithms [J]. Computer Science, 2018, 45(6A): 41-45, 62. (in Chinese)

[15] Carbonnel C, Cohen D A, Cooper M C, et al. On singleton arc consistency for CSPs defined by monotone patterns [J]. Computing Research Repository, 2018(96): 1-15.

[16] Kong S F, Lee J H, Li S J. A new distributed algorithm for efficient generalized arc-consistency propagation [J]. Autonomous Agents and Multi-Agent Systems, 2018, 32(5): 569-601.

[17] Lhomme O, Régin J C. A fast arc consistency algorithm for n-ary constraints [C]//Proceedings of the Twentieth National Conference on Artificial Intelligence and the Seventeenth Innovative Applications of Artificial Intelligence Conference, 2005: 405-410.

[18] Paredis J. Coevolutionary life-time learning [C]//Proceedings of International Conference on Parallel Problem Solving from Nature, Heidelberg: Springer, 1996: 72-80.

[19] Franz F, Paredis J, Mockel R. On the combination of coevolution and novelty search [EB/OL]. Piscataway, New Jersey: IEEE[2018-04-06]. <https://cris.maastrichtuniversity.nl/portal/files/25001423/Paper-crc.pdf>.