

文章编号: 1000-6788(2009)07-0097-08

面向任务的拼修策略问题及求解算法

李羚玮, 张建军, 张涛, 郭波

(国防科技大学 信息系统与管理学院, 长沙 410073)

摘要 研究了一种面向任务的拼修策略问题, 建立了该问题的多维背包问题模型, 提出了基于遗传算法的求解方案. 然后设计了一种求解效果较好的两阶段遗传算法, 进行了包括编码, 交叉, 变异, 最优前沿限定算法, 适应度函数, 选择策略和退火局部搜索算法在内的全面的分析和设计. 最后, 给出了一个算例, 检验了求解方案的实用性, 并且通过计算实验分析了遗传算法的有效性.

关键词 面向任务; 拼修策略; 多维背包问题; 两阶段遗传算法

中图分类号 TP307

文献标志码 A

Mission oriented cannibalization policy problem and its solving algorithm

LI Ling-wei, ZHANG Jian-jun, ZHANG Tao, GUO Bo

(School of Information System and Management, National University of Defense Technology, Changsha 410073, China)

Abstract A type of mission oriented cannibalization policy problem is studied this paper, the problem is modeled as a multi-dimensional knapsack problem, and a solution method what is based on genetic algorithm (GA) for the problem is presented. A two stages GA with better performance is designed and comprehensive analysis and design what including the optimal-frontier restriction algorithm, encoding, crossover, mutation, evaluation function, selection policy and simulated anneal based local search algorithm is performed. At last, the practicability of the solution method is verified by an example and analyzes the performance of the GA by computational experiment respectively.

Keywords mission oriented; cannibalization policy; multi-dimensional knapsack problem; two stages genetic algorithm

1 引言

拼修是在某装备产生故障的情况下, 使用原来属于其他装备上的外场可更换单元 (Line replaceable unit, LRU) 拼装出能够执行某种任务的装备的一系列活动. 拼修是装备修理的一种重要方式, 它能大大减少备件需求量, 或在有限备件资源条件下, 能够使尽量多的装备恢复战备完好, 从而提高装备的使用可用度.

传统的拼修策略是简单的, 即要求拼修出的装备上所有 LRU 都完好. 这种拼修策略实质上假设了所有装备面向的任务是单一的, 装备的功能是单一的, 并且装备的可靠性结构是由各个 LRU 串联的. 可是, 实际的装备和任务往往有以下特点: ①现代化复杂装备的功能往往有复合性, 即一种装备可能有多种功能, 且不同的功能对应不同的 LRU 组合 (LRU combination, LRUC); ②由于战场情况的复杂性, 装备面向的任务可

收稿日期: 2008-03-11

资助项目: 国家自然科学基金 (70501031)

作者简介: 李羚玮 (1982-), 广西桂林人, 博士研究生, 研究方向: 装备维修保障系统建模仿真与优化.

能有多个种类或多个阶段,且不同任务对应于不同的装备功能组合.可见,在不同的任务,多样的装备功能和有限的 LRU 之下,拼修的目标可能是恢复全部甚至部分功能^[1],因此拼修的策略也要随之改进.

基于上述理由,本文研究了多种任务下的复合功能装备的拼修策略问题,并建立了该问题的整数规划模型.在建模上本文依次开展了以下工作:①通过“任务→功能”的分解得到了任务关于装备功能的结构函数;②通过“功能→LRU”的分解得到了装备功能关于 LRU 的结构函数;③将以上两级的结构函数进行综合,得到了每种任务关于 LRU 的结构函数,并从中推导出每种任务所需要的 LRUC,再根据每种任务的重要度推导出对应的 LRUC 的重要度;④以拼修出的各种 LRUC 的数量为拼修策略,以各种 LRU 的数量为约束,以拼修策略和 LRUC 的重要度所确定的任务满足性评价函数为目标,建立了拼修策略问题的整数规划模型.

实际上,本文建立的拼修策略模型是一种多维背包问题(Multi-dimensional knapsack problem, MKP).MKP 是一类应用广泛、易于描述却难于求解的 NP 问题,在过去的 40 年中 MKP 得到了学术界的广泛关注并提出了大量算法.这些算法大致分为两类,一类是动态规划^[2],分支定界^[3]等传统算法,另一类是蚁群优化^[4]、遗传算法^[5]等近代优化算法.关于 MKP 的理论特性和求解算法的更详细的综述可见文献^[6].因为 MKP 的 NP 困难性导致其目前仍然没有有效的确定性算法,所以本文设计了一种两阶段遗传算法(Two-stages genetic algorithm, TGA)来求解本文建立的 MKP 模型.TGA 通过两个算法阶段的串联混合来达到兼顾进化初期的种群多样性和进化后期的收敛速度的目的,最后的计算试验表明了 TGA 的实用性和有效性.

本文的其余部分安排如下:第二节是问题描述与假设,第三节建立了优化模型,第四节设计了模型的求解算法,第五节给出了算例和仿真试验,第六节总结全文.

2 问题描述与假设

考虑一个拥有相同的复合型装备的装备群,对之作形式化描述如下:①该装备群可能执行的任务有 B 种,任务集合为 $M = \{M_1, M_2, \dots, M_b, \dots, M_B\}$, $b = 1, 2, \dots, B$.假设下一个任务 M' 是一个随机变量,其概率分布是 $\Pr\{M' = M_b\} = p_b$ 且 $\sum_{b=1}^B p_b = 1$;②该装备的功能有 E 种,功能集合为 $F = \{F_1, F_2, \dots, F_e, \dots, F_E\}$, $e = 1, 2, \dots, E$;③该装备的重要 LRU 有 A 种,LRU 集合为 $P = \{P_1, P_2, \dots, P_a, \dots, P_A\}$, $a = 1, 2, \dots, A$.假设由于故障或者战损导致 LRU 数量向量是 $\theta = (\theta_1, \theta_2, \dots, \theta_a, \dots, \theta_A)$,其中 θ_a 表示 P_a 的数量.

假设①任务 M_b 和装备功能集合 F 的关系函数为 α_b ,即 $M_b = \alpha_b(F)$;②功能 F_e 和 LRU 集合 P 的关系函数为 β_e ,即 $F_e = \beta_e(P)$.将 $F_e, e = 1, 2, \dots, E$ 的表达式代入 M_b 的表达式,则可得 M_b 和 LRU 集合 P 的关系函数为 $M_b = \varphi_b(P)$.其中 $\varphi_b(P) = \sum_{i=1}^{K_b} \varphi_{bi}(P)$, K_b 是 $\varphi_b(P)$ 之中的项数,每个 $\varphi_{bi}(P)$ 都具有 $\varphi_{bi}(P) = P_1^{c_1} P_2^{c_2} \dots P_a^{c_a} \dots P_A^{c_A}$ 的形式. $\varphi_b(P)$ 的含义是:任意 $i = 1, 2, \dots, K_b$ 对应的 $\varphi_{bi}(P)$ 所代表的 LRUC 都可以完成 M_b . $\varphi_{bi}(P)$ 的含义是:代表了一种能执行 M_b 的 LRUC,对于 $a = 1, 2, \dots, A$,该 LRUC 需要的 P_a 的数量是 c_a .

3 优化模型建立

定义 $\varphi_{bi}(P)$ 的重要度函数为 $IM(\varphi_{bi}(P)) = \sum_{b'=1}^B p_{b'} \times \varphi_{b'}(\varphi_{bi}(P))$,其中,如果 $\varphi_{bi}(P)$ 是 $\varphi_{b'}(P)$ 中的项则 $\varphi_{b'}(\varphi_{bi}(P)) = 1$,否则 $\varphi_{b'}(\varphi_{bi}(P)) = 0$.需要注意的是,对于 $b \neq b'$ 的 $\varphi_b(P)$ 和 $\varphi_{b'}(P)$ 中可能有相同的项,即存在同一种 LRUC 可以完成 M_b 和 $M_{b'}$ 两种任务(有可能更多).

以所有的 LRUC, $\varphi_{bi}(P)$, $b = 1, 2, \dots, B$, $i = 1, 2, \dots, K_b$ 为元素,构造一个无重复元素的向量 $\phi = (\phi_1, \phi_2, \dots, \phi_h, \dots, \phi_H)$, $h = 1, 2, \dots, H$,其中 $IM(\phi_h) \geq IM(\phi_{h+1})$.因为对于 $b \neq b'$ 的 $\varphi_b(P)$ 和 $\varphi_{b'}(P)$ 中可能有相同的项,所以 $H \leq \sum_{b=1}^B K_b$.

定义向量 $\rho = (\rho_1, \rho_2, \dots, \rho_h, \dots, \rho_H)$, ρ_h 是 ϕ_h 中的上标组成的向量,即,若 $\phi_h = P_1^{c_{h,1}} P_2^{c_{h,2}} \dots P_a^{c_{h,a}} \dots P_A^{c_{h,A}}$,则 $\rho_h = (c_{h,1}, c_{h,2}, \dots, c_{h,a}, \dots, c_{h,A})$,其中对于任意 $h = 1, 2, \dots, H$; $a = 1, 2, \dots, A$,都有 $c_{h,a} \in$

N, N 是自然数集合. ρ_h 和 ϕ_h 都代表了重要度排在第 h 位的一种 LRU, 记之为 $LRUC_h$.

定义①变量: 拼修方案向量 $n = (n_1, n_2, \dots, n_h, \dots, n_H), n_h \in N, n_h$ 表示拼修出的 $LRUC_h$ 的数量; ②约束: LRU 数量约束条件 $\sum_{h=1}^H \rho_h \times n_h \leq \theta$; ③目标: 记 n 的任务满足性评价函数为 $f(n) = \sum_{h=1}^H IM(\phi_h) \times n_h$. 则拼修策略问题的整数规划模型可以建立如下.

$$\begin{aligned} \max_n \quad & f(n) = \sum_{h=1}^H IM(\phi_h) \times n_h \\ \text{s.t.} \quad & \begin{cases} \sum_{h=1}^H \rho_h \times n_h \leq \theta \\ n_h \in N, h = 1, 2, \dots, H \end{cases} \end{aligned} \quad (1)$$

其中, 第一个约束条件表明 LRU 的数量限制在 θ 之内, 第二个约束表明这是整数规划. 通过该模型的形式可以发现它是一个多维背包问题 (MKP).

4 求解算法设计

MKP 是 NP 完全问题, 不存在有效的求解算法, 所以需要使用智能优化算法进行求解. 遗传算法是一类基于基因交叉变异机制的智能优化算法, 它具有全局进化搜索的特点. 以下本文设计一种两阶段遗传算法 (Two-stage genetic algorithm, TGA) 进行模型 (1) 的求解.

本节其余部分安排如下: 4.1 对两阶段遗传算法进行概述; 4.2 介绍编码, 交叉和变异等 TGA 的基本要素; 然后 4.3 揭示交叉和变异结果可能位于可行域外的问题, 并提出最优前沿限定算法来解决之; 4.4 是 TGA 的选择操作, 其中介绍了两个阶段的选择方法和 TGA 阶段转换的判断条件; 4.5 介绍了一种用来加速局部收敛的退火局部搜索技术.

4.1 两阶段遗传算法概述

遗传算法的基本要素是交叉, 变异和选择操作, 它的一个应用难点是遗传算法和实际问题的结构性差异, 该差异导致算法的基因操作产生的后代难以保持在实际问题的可行域. 遗传算法的收敛性和解的多样性是一对要取得折中的主要矛盾, 由此产生了以下四类工作: ①保持解的多样性的技术, 如小生境技术^[7]等; ②加速遗传算法的局部收敛的技术, 如退火算法^[8], 本地搜索^[9]等; ③加速全局收敛的技术, 如免疫算法^[10]; ④为了取得较好的折中, 经常采用的办法是使遗传算法具有适应性^[7]. 虽然遗传算法的应用研究成果和混合改进很多, 但是因为遗传算法本质上是随机搜索算法, 并且其应用效果常常取决于具体问题特性和算法思想的实现细节, 所以算法的实际性能需要计算实验的效果来确定.

本文提出的 TGA 的核心设计技术是将选择策略分成了两个阶段. TGA 的设计思想是: ①将第一阶段的结果种群作为第二阶段的初始种群, 来克服第二阶段算法对初值的依赖性; ②通过两个算法阶段的串联混合来达到兼顾进化初期的种群多样性和进化后期的收敛速度的目的. TGA 的总体结构如图 1 所示, 图中的 k 的含义是种群的代数, 其他参数的含义可见后文.

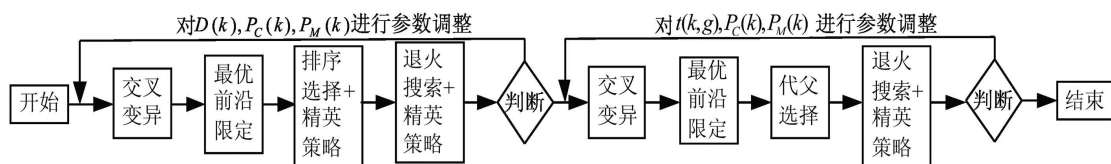


图 1 两阶段遗传算法设计

4.2 编码、交叉和变异操作实现

将可行解 $n = (n_1, n_2, \dots, n_h, \dots, n_H)$ 按照二进制编码方法转化成一个 H 维的基因串 $g = (g_1, g_2, \dots, g_h, \dots, g_H)$. 这样做的好处是使基因操作保持直观的解空间含义.

交叉操作使用按维交叉, 变异操作采用按位变异. 本算法采用参数适应性策略, 第 k 代种群的交叉概率和变异概率分别为 $P_C(k), P_M(k)$. 假设群体大小为 M_C , 第 k 代种群的个体集合是 $G(k)$. 经过交叉、变异

和最优前沿限定所产生的新种群 (未经过选择) 的个体集合是 $\widehat{G}(k)$, 则可以定义第 k 代种群交叉变异操作的报酬因子为 $\chi(k) = \frac{\sum_{g \in G(k)} f(g)}{\sum_{\widehat{g} \in \widehat{G}(k)} f(\widehat{g})}$, 则第 $k+1$ 代的交叉概率和变异概率分别为:

$$P_C(k+1) = P_C(k) \times \chi(k), \quad P_M(k+1) = P_M(k) \times \chi(k)$$

易知, 随着种群质量不断提高, 交叉变异产生更优解的可能性不断降低, 则 $\chi(k) \leq 1$, 那么 $\lim_{k \rightarrow \infty} P_C(k) = \lim_{k \rightarrow \infty} P_M(k) = 0$. 只要设定合适的参数初值, 这个参数控制策略可以在初期促进种群进化, 而在后期保护优良的基因特性.

4.3 最优前沿限定算法

遗传算法的交叉和变异结果不能够保证处于问题 (1) 的最优前沿上. 这包含两种情况: ①是 g 代表的分配方案超出 θ 的范围; ②是 g 的分配方案没有充分利用 θ 之中的所有 LRU. 为了使搜索过程保持在最优前沿, 本文采取了将基因操作结果修正到最优前沿上的办法, 其实质是在规划问题 (1) 中加入如下最优性约束:

$$\theta - \sum_{i=1}^H \rho_i \times n_i \geq \rho_h, \quad h = 1, 2, \dots, H \quad (2)$$

这个约束条件表明分配剩余的 LRU 数量不能拼出 $LRUC_h$, 它能将解限定在最优前沿上, 易知此约束是冗余的, 因为它实际上暗含于 (1) 的目标函数之中.

对应于上述第一种情况, 设计一种机会贪婪算法来将 g 修正到问题 (1) 的可行域, 成为 g' . 该算法的框图如图 2.

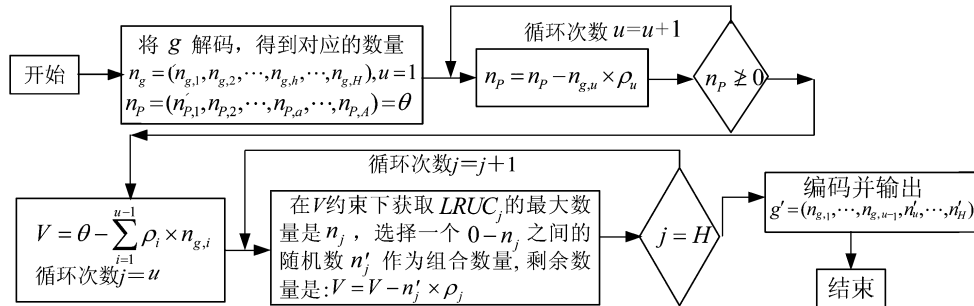


图 2 机会贪婪算法

如果处于问题 (1) 的可行域中的 g' 也满足式 (2) 约束, 则 g' 已经处于最优前沿. 否则可按照图 3 的贪婪算法将 g' 修正到规划问题 (1) 的最优前沿, 转化为 g'' .

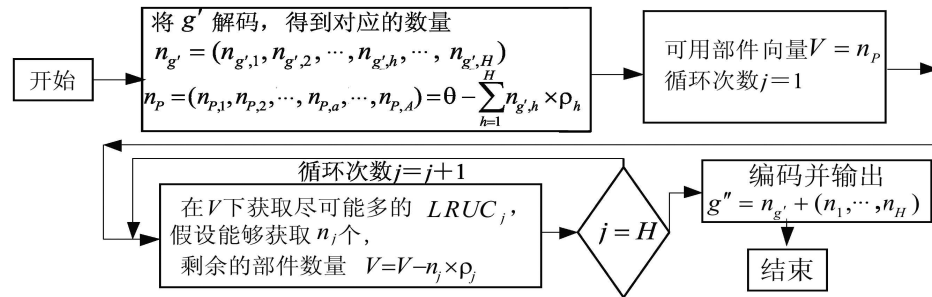


图 3 贪婪算法

实际上, 当 g 处于第一种情况时, 可直接用贪婪算法将 g 直接转化到最优前沿, 可是这会使遗传算法早熟收敛到用贪婪算法分配 θ 所得的方案上去. 机会贪婪算法能使修正算法有一定随机性, 从而使贪婪算法不至于起决定性作用. 但是, 最优前沿限定算法的随机性仍有可能破坏交叉变异所产生的基因特性, 使高强度的交叉变异接近于随机搜索.

4.4 适应度函数和选择策略

假设 $M'_C(k) = M_C + N_{CM}(k)$, $N_{CM}(k)$ 表示第 k 代种群交叉和变异产生的不同于父代的新基因数, 所以选择操作是在包含 $M'_C(k)$ 个个体的暂时种群 $\tilde{G}(k)$ 上进行的, $\tilde{G}(k) = G(k) \cup \hat{G}(k)$. 选择方案的要素有适应度函数和选择策略, 其设计目标是: 在进化早期保持种群的多样性; 后期加速收敛.

本文考虑了两种选择方案, 分别是: ①方案 1“排序选择 + 精英策略 + ψ ”; ②方案 2“以子代父 + f ”. 方案 1 是在保留精英的前提下, 直接对交叉变异之后的 $M'_C(k)$ 个个体按照 ψ 值排序并取前 M_C 个个体, 其中 ψ 是后文定义的评价函数; 方案 2 是子代优秀 (依照 f) 个体代替对应的父代个体. 表 1 是两者的比较.

表 1 两种选择方案的比较

名称	选择策略	评价函数	解的质量	多样性	进化速度	初值依赖性	收敛性
方案 1	排序选择 + 精英策略	ψ	振荡	较好	较快	较小	较差
方案 2	以子代父	f	平稳增长	较好	较慢	较大	较好

方案 1 中的 ψ 是一种考虑种群的多样性的评价函数, 对于 $\tilde{g} \in \tilde{G}(k)$, ψ 定义为 $\psi(\tilde{g}) = (d_{\tilde{g}})^{D(k)} \times f(\tilde{g})$. 其中 $(d_{\tilde{g}})^{D(k)}$ 是对 f 进行修正的距离因子, $d_{\tilde{g}} = \sum_{\tilde{g}' \in \tilde{G}(k)} d(\tilde{g}, \tilde{g}')$, $d(\tilde{g}, \tilde{g}')$ 是两个个体 \tilde{g} 和 \tilde{g}' 之间的欧氏距离, $D(k)$ 是用来对欧氏距离和 $d_{\tilde{g}}$ 进行控制的适应性参数. 假设 $\tilde{G}_f(k)$ 是将 $\tilde{G}(k)$ 按照 f 函数排序, 取最优 M_C 个产生的种群, 类似的, $\tilde{G}_\psi(k)$ 是将 $\tilde{G}(k)$ 先以 ψ 排序, 再取最优 M_C 个, 最后保留精英而产生的种群. 定义第 k 代种群按照方案 1 进行选择时的报酬因子为 $\zeta(k) = \frac{\sum_{\tilde{g}_\psi \in \tilde{G}_\psi(k)} f(\tilde{g}_\psi)}{\sum_{\tilde{g}_f \in \tilde{G}_f(k)} f(\tilde{g}_f)}$, 则可依据 $\zeta(k)$ 对 $D(k)$ 进行适应性控制, 即 $D(k+1) = D(k) \times \zeta(k)$. 由定义知 $0 \leq \zeta(k) \leq 1$, 所以 $D(k)$ 随 k 的增加而递减.

TGA 是和两种选择方案密切相关的: ①第一阶段采用方案 1; ②假设 \bar{D} 是设定的阈值, 若 $D(k+1) \leq \bar{D}$, 则开始第二阶段, 并且在该阶段采用方案 2. 第一阶段使用方案 1 的优势是可以优化种群的全局分布, 给第二阶段算法更好的初值; 第二阶段使用方案 2 的优势是可以保持优良种群的稳定性, 在最后也更易于算法收敛.

4.5 加速局部收敛的算法设计

本文设计一种退火局部搜索 (Simulated anneal local search, SALS) 算法来做每个个体的局部搜索. 对第 k 代种群 $G(k)$ 中某个个体 g 执行 SALS 的过程如图 4 所示, 其中 $t(k, g)$ 是 g 对应的初始温度, t_c 是使退火算法结束的门槛温度, Δt 是每次循环的降温值, g' 是算法的输出. 假设 n_g 是将 g 解码得到的拼修策略向量, $\tau(n_g)$ 是 n_g 在解空间中的邻域集合, $T(\tau(n_g))$ 是 SALS 中已经尝试过的 $\tau(n_g)$ 的元素.

SALS 的设计要点有: ①最初的 $t(1, g)$ 应当适当设小以避免搜索范围过宽, $g \in G(1)$. ②假设每次对 g 执行 SALS 完毕之后得到新解 g' , 则 $t(k+1, g') = t(k, g) \times f(g')/f(g) - \Delta t$, 如此可通过对 g 执行 SALS 的收益来调整 $t(k+1, g')$, 减掉常数 Δt 是为了使 SALS 的收敛性随着 k 的增加逐步增强. ③由于退火可能接受劣解, 所以对退火结果进行选择时需要保留精英 (可见图 1). SALS 如图 4 所示.

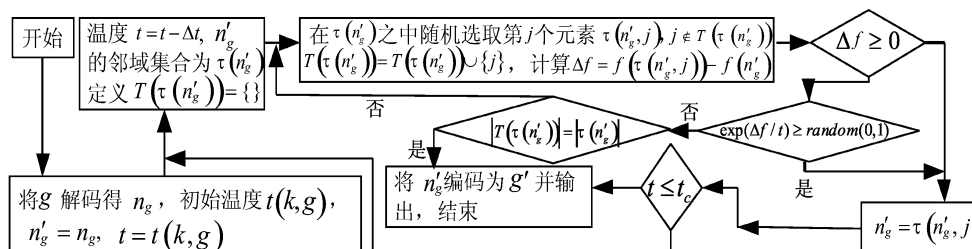


图 4 退火局部搜索算法

5 算例和仿真实验

5.1 算例

假定一个执行任务的装备群中有若干台同种复合型装备, 每种装备有 30 种 LRU, 分别是 P_1, P_2, \dots, P_{30} . 该装备包括 15 个基本功能, 分别为 F_1, F_2, \dots, F_{15} , 将功能分解到 LRU, 得到功能关系函数为:

$$\begin{aligned} F_1 &= P_1 P_2 + P_1 P_6 P_7, F_2 = P_1 P_4 P_{10}, F_3 = P_4 P_5 P_7, F_4 = P_7 P_9 + P_2 P_6, F_5 = P_3 P_{10}, F_6 = P_{23} + P_8, \\ F_7 &= P_{13} P_{19}, F_8 = P_{24} P_{25} + P_{21} P_{27}, F_9 = P_{11} P_{17} + P_{12} P_{14}, F_{10} = P_{19} P_{30}, F_{11} = P_{16} P_{18}, F_{12} = P_{15} P_{22}, \\ F_{13} &= P_{12} P_{17} P_{26} + P_{20} P_{28}, F_{14} = P_{17} P_{29} + P_6, F_{15} = P_{21} P_{25} P_{26} \end{aligned}$$

这个装备群可能的任务有 5 个, 分别为 $M = \{M_1, M_2, M_3, M_4, M_5\}$, 下阶段需要执行的任务为 M' , 其概率分布为 $\{0.3, 0.25, 0.2, 0.15, 0.1\}$. 将完成每个任务所需的能力分解到基本功能上, 得到任务关系函数为:

$$\begin{aligned} M_1 &= F_1 F_3 F_4 + F_2 F_9 F_{11}, M_2 = F_2 F_4 F_8 + F_{11} F_{15}, M_3 = F_5 F_7 F_{12} + F_{12} F_{14}, \\ M_4 &= F_6 F_9 F_{10} + F_5 F_{13}, M_5 = F_{12} F_{13} + F_8 F_{10} F_{11} \end{aligned}$$

将功能关系函数代入到任务关系函数, 得:

$$\begin{aligned} M_1 &= P_1 P_2 P_4 P_5 P_7 P_9 + P_1 P_4 P_5 P_6 P_7 P_9 + P_1 P_2 P_4 P_5 P_6 P_7 + \\ &P_1 P_4 P_{10} P_{11} P_{16} P_{17} P_{18} + P_1 P_4 P_{10} P_{12} P_{14} P_{16} P_{18} \\ M_2 &= P_1 P_4 P_7 P_9 P_{10} P_{24} P_{25} + P_1 P_4 P_7 P_9 P_{10} P_{21} P_{27} + P_1 P_2 P_4 P_6 P_{10} P_{24} P_{25} + \\ &P_1 P_2 P_4 P_6 P_{10} P_{21} P_{27} + P_{16} P_{18} P_{21} P_{25} P_{26} \\ M_3 &= P_3 P_{10} P_{13} P_{15} P_{19} P_{22} + P_{15} P_{17} P_{22} P_{29} + P_6 P_{15} P_{22} \\ M_4 &= P_{11} P_{17} P_{19} P_{23} P_{30} + P_{12} P_{14} P_{19} P_{23} P_{30} + P_8 P_{11} P_{17} P_{19} P_{30} + \\ &P_8 P_{12} P_{14} P_{19} P_{30} + P_3 P_{10} P_{12} P_{17} P_{26} + P_3 P_{10} P_{20} P_{28} \\ M_5 &= P_{12} P_{15} P_{17} P_{22} P_{26} + P_{15} P_{20} P_{22} P_{28} + P_{16} P_{18} P_{19} P_{24} P_{25} P_{30} + P_{16} P_{18} P_{19} P_{21} P_{27} P_{30} \end{aligned}$$

从上述结构函数推导出

$$\begin{aligned} \phi &= (P_1 P_2 P_4 P_5 P_7 P_9, P_1 P_4 P_5 P_6 P_7 P_9, P_1 P_2 P_4 P_5 P_6 P_7, P_1 P_4 P_{10} P_{11} P_{16} P_{17} P_{18}, P_1 P_4 P_{10} P_{12} P_{14} P_{16} P_{18}, \\ &P_1 P_4 P_7 P_9 P_{10} P_{24} P_{25}, P_1 P_4 P_7 P_9 P_{10} P_{21} P_{27}, P_1 P_2 P_4 P_6 P_{10} P_{24} P_{25}, P_1 P_2 P_4 P_6 P_{10} P_{21} P_{27}, P_{16} P_{18} P_{21} P_{25} P_{26}, \\ &P_3 P_{10} P_{13} P_{15} P_{19} P_{22}, P_{15} P_{17} P_{22} P_{29}, P_6 P_{15} P_{22}, \\ &P_{11} P_{17} P_{19} P_{23} P_{30}, P_{12} P_{14} P_{19} P_{23} P_{30}, P_8 P_{11} P_{17} P_{19} P_{30}, P_8 P_{12} P_{14} P_{19} P_{30}, P_3 P_{10} P_{12} P_{17} P_{26}, P_3 P_{10} P_{20} P_{28}, \\ &P_{12} P_{15} P_{17} P_{22} P_{26}, P_{15} P_{20} P_{22} P_{28}, P_{16} P_{18} P_{19} P_{24} P_{25} P_{30}, P_{16} P_{18} P_{19} P_{21} P_{27} P_{30}) \end{aligned}$$

假设装备群中 LRU 数量的总和是

$$\begin{aligned} \theta &= (24, 20, 28, 27, 28, 25, 23, 22, 27, 28, 25, 20, 28, 25, 21, 20, 27, 23, 28, \\ &23, 25, 27, 24, 29, 22, 25, 23, 22, 26, 24) \end{aligned}$$

依据 ϕ 容易得到 ρ , 由于篇幅限制, ρ 的描述在此省略. 将 ρ 和 θ 代入模型 (1) 可得到本算例的具体模型. 然后利用本文提出的两阶段遗传算法, 解得的最优方案中的一个是: $n = (19, 4, 0, 0, 0, 0, 0, 0, 1, 20, 0, 12, 9, 9, 15, 0, 0, 5, 22, 0, 0, 0, 0)$, 最优值为 $f(n) = 24.00$.

表 2 试验方案设计

项目方案	适应度函数	选择策略	SALS	参数自适应控制
A	$\psi(n) \rightarrow f(n)$	排序选择 + 精英策略 \rightarrow 代父选择	有	$t(k, g), P_C(k), P_M(k), D(k)$
B	$\psi(n)$	排序选择 + 精英策略	无	无
C	$\psi(n)$	排序选择 + 精英策略	无	$P_C(k), P_M(k), D(k)$
D	$f(n)$	代父选择	有	$t(k, g), P_C(k), P_M(k)$
E	$f(n)$	代父选择	无	$P_C(k), P_M(k)$
F	$f(n)$	排序选择	无	$P_C(k), P_M(k)$

5.2 算法试验分析

在此通过计算实验来对算法进行评价, 评价指标包括: 种群精英的平均适应度 \bar{f}^* , 个体间的平均距离 \bar{d} 和种群的平均适应度 \bar{f} , 其中 \bar{f}^* 和 \bar{f} 都基于评价函数 f . 各种实验方案的设计如表 2 所示, 其中两阶段遗传算法对应的是 A 方案.

设定种群规模为 100, 各实验 50 次, 在计算结果中取 100 代, 得到图 5, 6, 7 的结果.

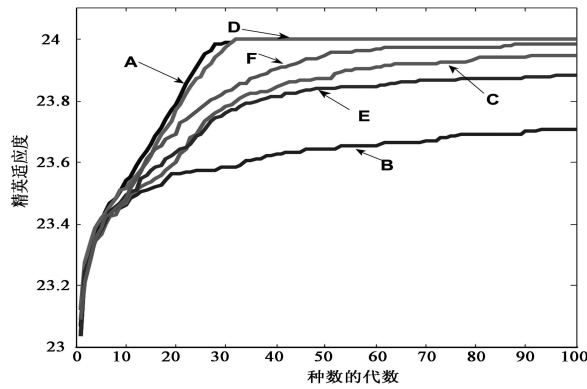


图 5 种群中最优值和种群代数的关系

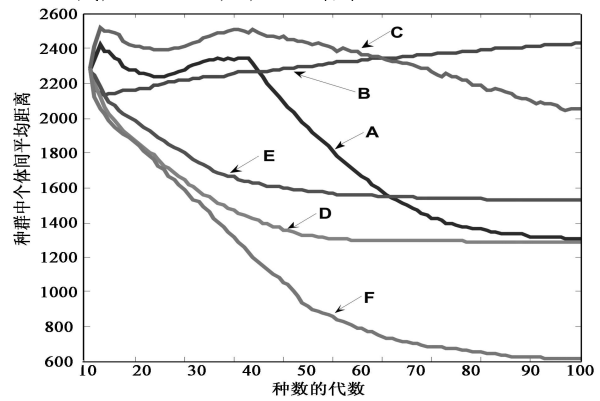


图 6 种群中平均距离和种群代数的关系

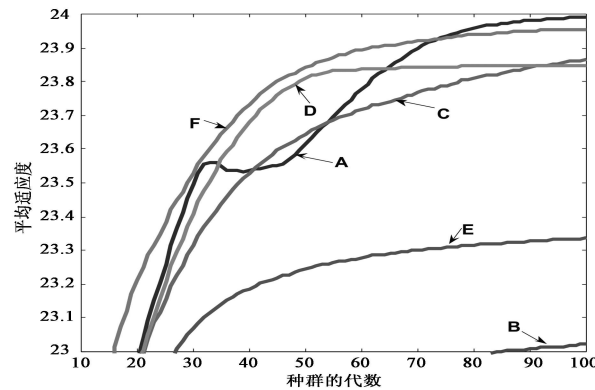


图 7 种群中平均适应度和种群代数的关系

图 5 展示的是种群中 \bar{f}^* 值和种群代数的关系. AD 方案的精英质量几乎一致且远优于其他的方案, 所以可知 SALS 对算法的局部寻优能力有决定性的影响. 而 B 和 C 两个方案的对比可知, 前者的寻优能力由于受到无控制的评价函数 ψ 的干扰而最差. E 和 F 的比较可以知道单纯代父选择寻找最优解的能力较弱, 这是初值依赖性造成的.

图 6 展示的是 \bar{d} 和种群代数的关系. AD 方案的对比说明了 A 方案第一阶段能够有效地增加种群的多样性. 本次实验将 A 的第一阶段限制在 30 代左右, 到 100 代时 A 算法的收敛程度已经基本和 D 相同. 而 B 方案表明“排序选择 + 精英策略 + ψ ”的选择方案确实能够保持解的多样性. E 和 F 两个方案的比较表明“代父选择”策略避免了种群的过早收敛.

图 7 展示的是 \bar{f} 和种群代数的关系. 当迭代 100 代时, A 方案比 F 方案的平均适应度略优, 可是结合图 6 可知 A 的种群分散程度远大于 F, 所以 A 方案解得的是可行域中更多的优解, 同时也证明了 F 方案是早收敛的. 再结合图 6、图 7 比较 A 方案和 D 方案可知: ① A 方案的第一阶段使种群的多样性得到加强和全局分布得到优化; ② A 方案的第二阶段算法虽然与 D 方案相同, 可是 A 方案中第二阶段算法的拥有较优初值, 因此 A 方案能在其 \bar{d} 参数始终优于 D 方案的情况下 \bar{f} 参数也迅速超越之; ③每次仿真中, A 方案的最终种群中的大多数个体都收敛到本问题的最优目标函数值 24 上, 取得了比较令人满意的效果.

6 结语

本文提出了一个面向任务的拼修策略问题, 然后建立了该问题的多维背包问题 (MKP) 模型, 提出了基

于遗传算法的求解策略. 接着根据求解 MKP 的需要设计了一种两阶段遗传算法, 最后通过算例求解和仿真实验分别表明了该算法的实用性和有效性.

参考文献

- [1] 甘茂治, 康建设, 高崎. 军用装备维修工程学 (第二版)[M]. 北京: 国防工业出版社, 2005.
Gan M Z, Kang J S, Gao Q. Military Equipments Maintainability Engineering(Second Edition)[M]. Beijing: National Defense Industry Press, 2005.
- [2] Baley S, Yanev N, Freville A, et al. A dynamic programming based reduction procedure for the multidimensional 0-1 knapsack problem[J]. European Journal of Operational Research, 2008, 186(1): 63-76.
- [3] Xia M, Stallaert J, Whinston A B. Solving the combinatorial double auction problem[J]. European Journal of Operational Research, 2005, 164(1): 239-251.
- [4] Kong M, Tianan P, Kao Y C. A new ant colony optimization algorithm for the multidimensional Knapsack problem[J]. Computers & Operations Research, 2008, 35(9): 2672-2683.
- [5] Joao Alvesa M, Almeida M. MOTGA: A multiobjective Tchebycheff based genetic algorithm for the multidimensional knapsack problem[J]. Computers & Operations Research, 2007, 34(11): 3458-3470.
- [6] Freville A. The multidimensional 0-1 knapsack problem: An overview[J]. European Journal of Operational Research, 2004, 155(1): 1-21.
- [7] 郭观七, 喻寿益, 贺素良. 自适应小生态遗传算法的理论分析和加速技术 [J]. 计算机学报, 2003, 26(6): 753-758.
Guo G Q, Yu S Y, He S L. Theoretic analysis and accelerating of a class of self-adaptive Niching genetic algorithms[J]. Chinese Journal of Computers, 2003, 26(6): 753-758.
- [8] 黄杰, 陈琳, 邹鹏. 一种求解极小诊断的遗传模拟退火算法 [J]. 软件学报, 2004, 15(9): 1345-1350.
Huang J, Chen L, Zou P. A Compounded genetic and simulated annealing algorithm for computing minimal diagnosis[J]. Journal of Software, 2004, 15(9): 1345-1350.
- [9] 叶慕静, 周根贵. 混合遗传算法在带走道的双目标布局问题中的应用 [J]. 系统工程理论与实践, 2005, 25(10): 101-107.
Ye M J, Zhou G G. The application of genetic algorithm in the bi-criteria layout problem with aisles[J]. Systems Engineering — Theory & Practice, 2005, 25(10): 101-107.
- [10] Jiao L C, Wang L. A novel genetic algorithm based on immunity[J]. IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans, 2000, 30(5): 552-561.