

改进型粒子群算法解决多维背包问题

郝春梅, 吴 波

(哈尔滨金融学院 计算机系, 黑龙江 哈尔滨 150030)

摘 要: 微粒群优化算法(PSO)是一种基于种群的随机优化技术. 将 EDA 算法与 PSO 算法结合起来, 形成一种新的改进的算法(EP SO). 算法将全局统计信息和全局最优解运用于解空间搜索, 以期能更有效解决组合优化问题, 最后将 EP SO 算法用于解决多维背包问题并进行了对比仿真实验. 实验结果表明, 在解决多维背包问题上, EP SO 优于传统的 PSO 算法以及多种启发式智能算法, 与此同时, EP SO 算法使用更少的参数, 因此更容易实现, 运行更加稳定, 效果更好.

关键词: 微粒群优化算法; 多维背包; 组合优化

中图分类号: TP393

文献标识码: A

文章编号: 1000-7180(2012)09-0129-04

Improved Particle Swarm Algorithm for the Multi-dimensional Knapsack Problem

HAO Chun-mei, WU Bo

(Department of Computer Science, Harbin Finance University, Harbin 150030, China)

Abstract: Particle Swarm Optimization algorithm (PSO) is a population-based stochastic optimization techniques. In this paper, the EDA algorithm and PSO algorithm combine to form a new improved algorithm (EP SO). The EP SO algorithms can effectively global statistics and global optimal solution used in the solution space search, the EP SO algorithm for solving the multidimensional knapsack problem. Experimental results show that solving the multidimensional knapsack problem, the EP SO is better than the traditional PSO algorithm, as well as a variety of heuristic intelligent algorithms, and at the same time, the EP SO Algorithm using fewer parameters, and therefore easy to implement, run more stable.

Key words: particle swarm optimization algorithm; multidimensional knapsack; combination optimization

1 引言

近年来进化计算方法发展迅速, 已成为了当前研究的热点, 并运用到智能计算的各个领域. 进化计算可将 NP 难问题转为在多项式时间内求解最优问题来求解^[1]. 微粒群算法(Particle swarm optimization algorithm, PSO)是其中有代表性的进化计算. 微粒群算法(PSO)是在 1995 年由美国社会心理学家 James Kennedy 和电气工程师 Russell Eberhart 共同提出的, 其基本思想是受他们早起对鸟类群体行为研究结果的启发, 并利用了生物学家 Frank

Heppner 的生物群体模型^[2]. 微粒群优化算法(PSO)是一种基于种群的随机优化技术. 分布估计算法是进化计算领域新兴起的一类随机优化算法. 分布估计算法是遗传算法和统计学习的结合, 通过统计学习的手段建立解空间内个体分布的概率模型, 然后对概率模型随机采样产生新的群体, 如此反复进行, 实现群体的进化^[3].

本文将分布估计概率算法与粒子群算法有效的结合在一起, 形成了一个混合的离散粒子群算法(EP SO). 使得算法能很好地将全局统计信息与全局最优解结合起来, 以期能更有效解决组合优化问

收稿日期: 2012-02-10; **修回日期:** 2012-03-20

基金项目: 黑龙江省教育厅科学技术研究项目(12511065)

题以及 NP 难问题.

2 概述

2.1 微粒群算法(PSO)

微粒群算法(PSO)来源于对一个简化社会模型的模拟^[4]. 研究人员通过对动物社会群体行为的观察,发现在群体中对信息的社会共享有利于在演化中获得优势,把这些作为研究 PSO 算法的基础. 通过消除不必要的变量,加入邻近的速度匹配,并考虑多维搜索以及根据距离的加速,最初版本 PSO 就形成了^[5].

$$v_{id} = wv_{id} + c_1 \text{rand}() (p_{id} - x_{id}) + c_2 \text{Rand}() (p_{gd} - x_{id}) \quad (1)$$

$$x_{id} = x_{id} + v_{id} \quad (2)$$

其中 c_1, c_2 为加速常数, w 为惯性权重, $\text{rand}()$ 和 $\text{Rand}()$ 为两个在 $[0, 1]$ 范围内变化的随机函数. 粒子的速度 v_i 被一个最大速度 $v_{\max}$ 所限制.

2.2 分布估计算法

分布估计算法(Estimation of distribution algorithm, EDA)是一类新型的优化算法,目前迅速成为进化计算领域的研究热点和解决工程问题的有效办法^[6]. 分布估计算法的概念最早在 1996 年提出,在 2000 年前后迅速发展,成为当前进化计算领域前沿的研究内容^[7]. 分布估计算法提出了一种全新的进化模式. 分布估计算法是概率模型的学习和采样. 分布估计算法通过一个概率模型描述候选解在空间的分布,采用统计学习手段才能够群体宏观的角度建立一个描述解分布的概率模型,然后对概率模型随机采样产生新的种群,如此反复进行,实现种群的进化,直到终止条件.

3 改进型粒子群算法求解多维背包问题

3.1 修复算子

MKP 问题中有 N 个物体 M 个背包,每个背包都有一个容量上限 $c_j (j=1, \dots, m)$. 用一个向量 $x_i (i=1, \dots, n)$ 表示背包的装入情况, x_i 为 1 表示装入该物体,为 0 表示不装入该物体. 每个物体有一个价值 $p_i (i=1, \dots, n)$ 和一个权重 w_{ij} . MKP 问题的目标是在不超过背包容量的情况下,使得背包内物体的总价值最大^[8-9]. 因此,在实现时,每个粒子数据结构中含有一个向量 $\text{number}[\text{DIM}]$ 表示物体的选择情况. 这样,每个粒子就是背包问题的一个解. $\text{number}[\text{DIM}]$ 向量的长度等于物体的个数 N ,相当于在 N -DIMN 空间的搜索. 向量如下:

$$\text{number} = [x_1, x_2, \dots, x_n] \quad (3)$$

但是这种表示方法在搜索时可能产生不可行解即解可能违反背包约束,形如:

$$\sum_{j=1}^n w_{ij} \text{number}[j] > c_j, \quad i = 1, \dots, m \quad (4)$$

此时,应对不可行解进行修复. 修复方法根据物品的伪效用比 u ,利用贪婪算法对不可行解进行修复. 伪效用比类似于单背包问题的价值密度,在单背包问题中物品 j 的价值密度等于物品的价值 p_j 与物品所占的资源 w_j 的比值 p_j/w_j ,价值密度的值越大,说明物品越好,被选择的机会就越大. 对于多维背包问题,最简单的伪效用比是根据单背包问题的价值密度得出的,即物品 j 的价值 p_j 与物品相对于不同约束所占的资源之和 $\sum_{i=1}^m r_{ij}$ 的比值 $(p_j/\sum_{i=1}^m r_{ij})$. 修复算子如下:

$$u_j = \frac{p_j}{\sum_{i=1}^m r_{ij} w_{ij}} \quad (5)$$

$$r_i = \frac{\sum_{j=1}^n w_{ij} - c_i}{\sum_{j=1}^n w_{ij}} \quad (6)$$

在获得 u 之后,对 u 按从大到小进行排序,修复算子根据 u 的值考虑是否把物品拿出或加入. 第一步操作(取出):如果新个体为不可行解或当前种群中存在与之相同的个体,则把包中 u 值最小的物品拿出,如果该个体还是不可行解或当前种群中有相同的个体,则重复这个操作,直到该个体为可行解且当前种群中没有与之相同的个体为止. 第二步操作(加入),在第一步得到可行解后,按 u 值从大到小考虑所有不在包中的物品,寻找是否存在这样的物品,即加入包后该个体仍为可行解且当前种群中没有与之相同的个体,有则加入.

3.2 适应函数设置

由于背包问题的特殊性,可以将适应函数定义为 $f(x) = \sum_{i=1}^n p_i x_i$,其中 p_i 为每个物体的价值.

$$x_i = \begin{cases} 1 & \text{物体被选中} \\ 0 & \text{物体没被选中} \end{cases}$$

对背包的选择要满足 $\sum_{i=1}^n w_i x_i \leq C$,其中 w_i 为每个物体的权重, C 为每一种资源的上限. 实验中将随机选择参数 β 设置为 0.98.

3.3 UMD 算法

实验采用 UMD 算法概率模型. UMD 算法由

德国学者 Muhlenbein 在 1996 年提出,采用 UMD 算法流程如下描述如下:

- (1) 随机产生 POPSIZE 个个体作为初始群体 $D_l, l=0$,其中 POPSIZE 为群的大小;
- (2) 计算 POPSIZE 个个体的适应值,如果符合终止条件,算法结束,否则继续进行;
- (3) 进行选择操作,选择 $N < M$ 个个体作为优势群体 D_i ,即适应值最大者;
- (4) 由优势群体 D_i (即当前最优解集合 Local best)构建概率模型,估计联合概率分布

$$p_i(x) = p(x | D_i) = \prod_{i=1}^n p_i(x_i)$$
$$= \prod_{i=1}^n \frac{\sum_{j=1}^N \delta_j(X_i = x_i | D_i)}{n}$$

(7)

其中 n 为问题维数(变量个数),即物体的个数.

$$\delta_j(X_i = x_i | D_i) = \begin{cases} 1 & X_i = x_i \\ 0 & \text{其他} \end{cases}$$

即使用当前最优(Locate best)的 N 个解向量 number[]生成下一代的概率向量 $p_i(x)$.

- (5) 从概率向量 $p_i(x)$ 中采样 POPSIZE 次,得到新一代大小为 POPSIZE 的群体,返回(2). 具体实现时,运行 500 代结束.

4 实验结果与分析

实验使用的测试实例是由 Beasley 和 Chu 所提供的 270 个多维背包问题. 其中约束个数 DIM 包括 5、10 和 30,物品数量 BAG 包括 100、250 和 500,每一组 DIM-BAG 各有 30 个问题. 这 270 个实例的生成方法如下:物品 j 消耗资源 i 的量 w_{ij} 为一个均匀分布在区间 $(0, 1000)$ 上的整数. 对于每一个 DIM-BAG 的组合,每个资源约束 $b_i = \alpha \sum_{j=1}^n w_{ij}$, α 是问题的紧密比,前十个问题的 α 取 0.25,接下来十个问题的 α 取 0.50,最后十个问题的 α 取 0.75. 由于很多问题的最优解还不知道,所以采用 100(松弛 LP 最优解—所求最优解)/(松弛 LP 最优解)的值对所求解的评估,记为 %pa. 显然,%pa 越小,该解就越接近最优解.

(1) EPSO 计算时间统计

测试时,粒子群规模 POPSIZE 取 50,随机运行 50 次,每次运行 500 代. 表 1 是根据不同的 DIM, BAG 和 α 对 MKP 的计算时间进行统计. 从表 1 可以看出使用 EPSO 算法可以高效地找出 MKP 问题的最优解. 随着问题规模的增大,EPSO 算法的优点

更加体现出来. 其中 DIM 为体的个数,BAG 为背包个数,即约束个数. α 为问题的紧密比,a. z. t 表示 EPSO 找到的最好解所花的平均时间,a. s. t 表示 EPSO 计算所用的平均时间(单位:秒).

表 1 MKP 计算时间统计表

问题			a. z. t	a. s. t
DIM	BAGN	α		
5	100	0.25	7.21	132.77
		0.50	11.35	130.73
		0.75	4.27	122.34
5	250	0.25	116.82	281.88
		0.50	166.93	278.68
		0.75	142.31	301.20
10	100	0.25	16.17	118.97
		0.50	22.21	112.86
		0.75	7.69	124.34
5	500	0.25	437.11	567.13
		0.50	303.19	491.49
		0.75	367.55	619.86
10	250	0.25	138.52	283.87
		0.50	137.32	286.30
		0.75	135.36	273.81
10	500	0.25	292.00	527.55
		0.50	279.38	431.66
		0.75	334.12	509.03
30	100	0.25	32.92	132.79
		0.50	43.50	160.10
		0.75	22.45	141.25
30	250	0.25	164.45	323.35
		0.50	207.43	331.13
		0.75	110.25	334.17
30	500	0.25	395.86	557.04
		0.50	380.26	607.81
		0.75	387.51	612.07
总平均值			172.84	325.93

(2) EPSO 与 PSO 比较

表 2 将 EPSO 算法与 PSO 算法的计算结果进行比较. 从表 2 可以看出,对物品数 n 小于 500 的实例的求解,EPSO 要略优于 PSO. 由于 5-100 这组实例由于规模相对较小,EPSO 和 PSO 都能找到全部最优解;对于 5-250 和 10-100 这两组实例,EPSO 找到的最优解都比 PSO 多,好于 PSO;而对于其它三组实例,虽然不能确定 EPSO 找到的就是最优解,但从 %pa 的对比可以看出,EPSO 的 %pa 小于 PSO 的 %pa,说明 EPSO 的解更接近于最优解. 对物品数 n 等于 500 这三组实例的求解,EPSO 则不如 PSO,特别是 30-500 这组实例,EPSO 与 PSO 的差距相对于其它各组大,对于 5-500 这组实例,EPSO 的平均 %pa 略高于 PSO,而对于 10-500 这组实例,EPSO 和 PSO 的平均 %pa 一样.

表 2 EP SO 与 PS O 的计算结果对比表

问题			PSO		EPSO	
DIM	BAG	α	平均 %pa	最优 解	平均 %pa	最优 解
5	100	0.25	0.979	10	0.990	10
		0.50	0.451	10	0.452	10
		0.75	0.358	10	0.319	10
		平均值		0.596		0.587
5	250	0.25	0.227	8	0.226	9
		0.50	0.126	5	0.113	6
		0.75	0.089	5	0.077	9
		平均值		0.141		0.138
10	100	0.25	1.562	10	1.563	10
		0.50	0.755	9	0.791	10
		0.75	0.482	10	0.482	10
		平均值		0.956		0.946
5	500	0.25	0.084	n/k	0.092	n/k
		0.50	0.042	n/k	0.043	n/k
		0.75	0.024	n/k	0.027	n/k
		平均值		0.051		0.054
10	250	0.25	0.501	n/k	0.491	n/k
		0.50	0.252	n/k	0.237	n/k
		0.75	0.153	n/k	0.149	n/k
		平均值		0.301		0.293
10	500	0.25	0.228	n/k	0.225	n/k
		0.50	0.105	n/k	0.111	n/k
		0.75	0.073	n/k	0.069	n/k
		平均值		0.136		0.136
30	100	0.25	2.915	n/k	2.914	n/k
		0.50	1.342	n/k	1.331	n/k
		0.75	0.829	n/k	0.829	n/k
		平均值		1.696		1.692
30	250	0.25	1.179	n/k	1.166	n/k
		0.50	0.521	n/k	0.518	n/k
		0.75	0.305	n/k	0.302	n/k
		平均值		0.669		0.663
30	500	0.25	0.613	n/k	0.689	n/k
		0.50	0.264	n/k	0.299	n/k
		0.75	0.166	n/k	0.174	n/k
		平均值		0.348		0.388
总平均值			0.542		0.545	

综上所述,EP SO 对大规模背包问题的求解要略优于 PS O,而对于超大规模的背包问题,虽然 EP SO 的平均 %pa 比 PS O 的大 0.002,但在部分实例的求解中,EP SO 能找到比 PS O 好的解,可以说 EP SO 整体性能要优于 PS O.

5 结束语

本文将分布估计概率算法与粒子群算法有效的结合在一起,形成了一个混合的离散粒子群算法 (EP SO). EP SO 算法能很好地将全局统计信息与全局最优解结合起来,使用 EP SO 算法解多维背包问题,借以测试 EP SO 算法的解决问题的能力. 测试结果表明,EP SO 算法比传统的 EDA 算法,PS O 算法的解题能力都要强. 与此同时,EP SO 算法使用更少的控制参数,使得该算法更容易实现,运行更稳定.

参考文献:

- [1] 蔡自兴. 人工智能及其应用:研究生用书(第三版)[M]. 北京:清华大学出版社,2005.
- [2] Jiahai Wang, A novel discrete particle swarm optimization based on estimation of distribution[C]// Proceedings of the 3rd International Conference on Intelligent Computing, Qingdao: ICIC, 2007: 791-802.
- [3] 周树德,孙增圻. 分布估计算法综述[J]. 自动化学报,2007,33(2): 113-124.
- [4] 任子晖,王坚. 动态拓扑结构的多目标粒子群优化算法[J]. 同济大学学报:自然科学版,2011,39(8): 1222-1226.
- [5] Zhou Chi, Zhang Xuejun, CAI Kaiquan, et al. Comprehensive learning multi-objective particle swarm optimizer for crossing waypoints location in air route network[J]. Chinese of Journal Electronics, 2011, 20(3): 533-538.
- [6] 周雅兰,王甲海,印鉴. 一种基于分布估计的离散粒子群优化算法[J]. 电子学报,2008,36(6): 1242-1248.
- [7] 何小娟,曾建潮. 基于优良模式连接的分布估计算法求解 TSP 问题[J]. 模式识别与人工智能,2011,24(2): 185-193.
- [8] 喻学才,张田文. 多维背包问题的一个蚁群优化算法[J]. 计算机学报,2008,31(5): 810-819.
- [9] 李羚玮,张建军,张涛,等. 面向任务的拼修策略问题及求解算法[J]. 系统工程理论与实践,2009,7(3): 97-104.

作者简介:

郝春梅 女,(1975-),副教授,硕士. 研究方向为管理信息系统.

吴波 男,(1972-),副教授,硕士. 研究方向为电子商务.