

一种求解多维背包问题的小世界算法

杜巍¹, 李树茁², 陈煜聪³

(1. 西安交通大学管理学院, 710049, 西安; 2. 西安交通大学人口与发展研究所, 710049, 西安;

3. 西安交通大学机械工程学院, 710049, 西安)

摘要: 针对遗传算法求解复杂组合优化问题时出现早熟收敛和种群多样性丧失等问题,提出了一种解决多维背包问题的二进制编码小世界算法(BSWA). BSWA 算法依据社会学中的小世界现象搜索机理,采用类似遗传变异操作的局部搜索,而非遗传算法中的交叉操作. 针对多维背包问题的多约束性,BSWA 算法还按照价值资源比大小对不可行解进行贪婪修正,以保证求解的正确性. 与遗传算法相比,BSWA 可以在一定程度上克服早熟收敛,在保持种群多样性和求解精度方面均体现出较大的优势,具有解决复杂组合优化问题的潜力. 对 55 个标准的多约束 0-1 背包问题进行了 50 次随机实验,结果表明,BSWA 算法对于其中 72.73% 的问题可以次次获得最优解,对于其他不能次次求解到最优解的问题,也可以获得非常接近全局最优解的满意解.

关键词: 小世界算法; 多维背包问题; 贪婪修正算子

中图分类号: TP183 **文献标志码:** A **文章编号:** 0253-987X(2009)02-0010-05

A Small World Algorithm for Multi-Dimensional Knapsack Problems

DU Wei¹, LI Shuzhuo², CHEN Yucong³

(1. School of Management, Xi'an Jiaotong University, Xi'an 710049, China; 2. Institute for Population and

Development Studies, Xi'an Jiaotong University, Xi'an 710049, China; 3. School of Mechanical

Engineering, Xi'an Jiaotong University, Xi'an 710049, China)

Abstract: In order to overcome the shortcomings of genetic algorithms (GA), an optimization algorithm called the binary-coding small world algorithm (BSWA) is proposed. The GA always loses diversity in the set of the candidate solutions and prematurely converges when it is used to solve complex combinatorial optimization problems. The BSWA is based on the searching mechanisms in social networks, and emphasizes local (as mutation in GA) rather than global search (as crossover in GA) to find solutions for optimization problems. Compared with the GA, the BSWA is capable of preserving diversity and avoiding premature convergence, and converges faster. These properties suggest that the BSWA is a useful method for solving complicated optimization problems. Simulation results show that the best known solutions of 72.73% of the 55 standard 0-1 knapsack problems can be found by the BSWA in each of the 50 independent runs, and the final solutions found by the BSWA for the other problems are very close to the best known ones.

Keywords: small world algorithm; multidimensional knapsack problem; greedy correcting operator

背包问题(KP)是运筹学中常见的 NP 完全问题,在许多工业实践中有着重要的应用^[1]. 求解背包

问题的传统精确算法包括枚举法、图论法和动态规划法等^[2],这类方法虽然可以保证求得最优解,但是

计算复杂度高,对于大规模多维背包问题求解速度慢。贪婪算法、模拟退火算法和禁忌搜索算法等近似算法虽运算速度快,但是算子设计需要一定的经验和技巧,算法稳定性差,且不能保证求解到最优解^[3]。遗传算法、蚁群算法和人工免疫算法等智能算法也逐渐用于解决背包问题等组合优化问题^[4]。与传统优化算法相比,智能优化算法具有并行处理、全局搜索和局域搜索能力,且算法的快速性、鲁棒性好,表现出一定的解决复杂问题的潜力。

遗传算法在求解背包问题等复杂组合优化问题时,往往会出现早熟收敛和种群多样性丧失快等问题^[1],而小世界算法在一定程度上可以克服遗传算法的上述缺陷,在求解复杂问题时表现出一定优势^[5]。目前将小世界算法用于解决组合优化问题的成果还不多见,本文的主要工作就是尝试采用小世界算法来解决多维背包问题(MKP)。

1 背包问题的数学描述

给定背包的 m 种资源,如尺寸、承重等,其中第 i 种资源取值为 $c_i, i=1,2,\dots,m$;另给定 n 件物品,其中第 j 件物品的价值为 p_j ,对应资源 i 的占用量为 $a_{ij}, j=1,2,\dots,n$ 。受背包各种资源限制,若只能挑选一部分物品装入背包,那么如何从 n 件物品中选出部分物品装入背包使装入背包中的物品价值最大,其数学描述为

$$\left. \begin{aligned} \max f(\mathbf{x}) &= \sum_{j=1}^n x_j p_j \quad j=1,2,\dots,n \\ \text{s. t.} \quad \sum_{j=1}^n x_j a_{ij} &\leq c_i \quad i=1,2,\dots,m \\ \mathbf{x} &= \{x_1, x_2, \dots, x_n\} \quad x_j \in \{0,1\} \end{aligned} \right\} \quad (1)$$

式中: $f(\mathbf{x})$ 代表装入背包物品的总价值;参数 $c_i > 0, p_j > 0, 0 < a_{ij} < c_i$; $\mathbf{x} = (x_1, x_2, \dots, x_n)$ 表示一组物品的选择策略, $x_j = 1$ 表示物品 j 被选中放入背包, $x_j = 0$ 表示物品 j 没有被挑选。当参数 m 取值为 1 时,问题变成为简单的 0-1 背包问题,当 $m \geq 2$ 时,称为多维 0-1 背包问题。

2 算法设计与算法流程

2.1 编码策略与适应度函数

根据背包问题的特点,本文采用二进制编码的小世界算法(BSWA)来求解 MKP^[5],节点 $s_i = \{s_1, s_2, \dots, s_l\}$,其中 $s_j \in \{0,1\}, j=1,2,\dots,l$ 依次表示物品的序号, s_j 取 1 表示第 j 个物品被选中装入背包,

取 0 表示没有物品被选中。

背包问题以效益最大化为目标,即 $\max f(\mathbf{x}) =$

$\sum_{j=1}^n x_j p_j, j=1,2,\dots,n$,本文以此作为算法的适应度函数。

2.2 算子设计

BSWA 的算子主要包括局部搜索算子 Ψ 和随机长连接搜索算子 Γ ,二进制编码算法中 2 个算子的具体实施在此不再赘述,详见文献^[5]。

由于约束条件的存在,随机产生的初始解以及经由算子 Ψ 和 Γ 的操作而产生的备选解不一定为可行解,因此,在进行解码操作时要首先对每个原始节点 s_i 进行可行性判断,并对不可行解进行修正使之成为可行解。所谓的不可行解是指,对所有的 $i=1,2,\dots,m$,不能保证条件 $\sum_{j=1}^n s_j a_{ij} \leq c_i$ 均成立。对于不可行解,在文中采用贪婪修正策略^[6],具体操作如下:如果被选物品超出资源 i 的最大容量 c_i ,则按照价值资源比 $\frac{p_i}{a_{ij}}$ 的大小,从被选物品中取出 $\frac{p_i}{a_{ij}}$ 最小的物品 j ,并将其在节点 s_i 中的对应编码 s_j 由 1 变为 0,如果此时的解仍不能满足所有约束条件,则按照前面的修正原则,继续从背包中剔除物品,直到该解成为可行解。

2.3 算法流程

f^* 表示背包问题已知最优解, $S'(k) = \{s'_1(k), s'_2(k), \dots, s'_n(k)\}$ 表示修正后的节点集合, p_s 表示邻域搜索概率, $S_i(k+1)$ 表示节点 i 在第 k 代进行 SWA 搜索所得备选节点集合, $e^{-1}(s)$ 表示对节点 s 进行解码操作, $f(e^{-1}(s))$ 表示节点解码后的适应度值, n_c 表示备选节点集合 $S_i(k+1)$ 中的节点数, $k_{\max}$ 表示最大允许迭代次数。

BSWA 算法求解背包问题的流程如下。

(1)生成初始节点集合 $S(0) = \{s_1(0), s_2(0), \dots, s_n(0)\}$ 。

(2)对集合 $S(0)$ 中的每一个节点进行可行性判断并修正。

(3)计算修正后的 $S'(0)$ 中各节点的适应度值。

(4)判断初始群体中最优解 $f^*(0)$ 是否满足终止条件 $f^*(0) = f^*$,如满足,则程序终止,否则转至步骤(5)。

(5)小世界搜索迭代,直到满足终止条件为止;

(5-1)初始化迭代次数 $k=0$;

(5-2)节点集合 $S'(k) = \{s'_1(k), s'_2(k), \dots,$

$s'_n(k)\}$ 进行小世界搜索;

(5-2-1)初始化参数 $i=0$;

(5-2-2)单个节点进行小世界搜索;

(5-2-2-1)初始化参数 $j=0$;

(5-2-2-2)生成 0 到 1 之间的随机数 p_r , 如果 $p_r < p_s$, 则转至(5-2-2-3), 否则转至步骤(5-2-2-4);

(5-2-2-3)在算子 Ψ 的作用下进行局部短连接搜索, 即 $\dot{s}_i(k+1) \leftarrow \Psi(s'_i(k))$, 转至步骤(5-2-2-5);

(5-2-2-4)在算子 Γ 的作用下进行随机长连接搜索, 即 $\ddot{s}_i(k+1) \leftarrow \Gamma(s'_i(k))$, 转至步骤(5-2-2-5);

(5-2-2-5)将搜索到的节点 $\dot{s}_i(k+1)$ 或 $\ddot{s}_i(k+1)$ 存放在备选节点集合 $S_i(k+1)$ 中, $j=j+1$;

(5-2-2-6)循环结束判断, 若 $j=n_c$ 则转至步骤(5-2-2-7), 否则转至步骤(5-2-2-2);

(5-2-2-7)对备选节点集合 $S_i(k+1)$ 中的每个节点进行可行性判断, 并对不可行解进行贪婪修正;

(5-2-2-8)计算修正后的被选节点集合 $S'_i(k+1)$ 中各个节点的适应度值, 并从中选取适应度值最大的一个 $s'_i(k+1)$, 若 $f(e^{-1}(s'_i(k+1))) > f(e^{-1}(s'_i(k)))$, 则 $s'_i(k+1)$ 代替原节点 $s'_i(k)$ 进入下一代循环, 否则 $s'_i(k)$ 保留原值进入下一代循环;

(5-2-2-9) $i=i+1$;

(5-2-2-10)循环结束判断, 如果 $i \leq n$ 则转至步骤(5-2-2), 否则转至步骤(5-3);

(5-3)计算新一代节点 $S'(k+1) = \{s'_1(k+1), s'_2(k+1), \dots, s'_n(k+1)\}$ 的适应度值, 同时给出最优节点 $s^*(k+1)$ 的适应度值 $f^*(k+1)$;

(5-4) $k=k+1$;

(5-5)如果终止条件满足, 则程序结束, 否则转至步骤(5-2).

实践中一般采用限定目标函数计算次数(有时也采用迭代次数), 或在连续几次(如 t 次)迭代中节点种群的最好解都无法改善, 或者二者的混合形式作为终止条件. 由于本文实验测试问题的最优值已知, 所以采用 $k > k_{\max}$ 和 $f^*(k+1) = f^*$ 中的任意一个成立作为算法终止条件.

3 实验与讨论

实验 1 实验数据取自文献[7], 包括 6 类共 55 个

标准背包问题, 物品数在 15 到 105 之间, 背包数在 2 到 30 之间, 各个问题最优解已知. 实验中节点数 n 取 40, n_c 取 4, 与算子 Ψ 相关的参数邻域半径 l 取为 1, p_s 取 0.8. 实验中算法终止的判断标准是: 如果当前函数值评价次数超出最大允许值 2×10^5 或者当前最优解为该背包问题的全局最优解, 则程序终止. 其中, 计算一次单个节点的适应度值即为一次函数值评价. BSWA 算法对每个问题均独立运行 50 次, 并对最优解的次数 O_n 和平均最优函数值 f_{mean} 进行了统计, 结果列于表 1 中. 表 1 同时还列出了各个背包问题的名称、物品数 n 、限制资源数 m 以及各个背包问题的已知最优解 f^* .

结果表明, 对于实验中的大多数标准背包问题, BSWA 算法通过较少的函数值评价次数求解到全局最优解. 其中, 72.73% 的背包问题在 50 次独立运行实验中能够次次求解到最优解. 除问题 SENT1 和 SENT2 外, 对于其他不能次次求解到全局最优解的情况, BSWA 算法可以求解到非常接近全局最优解的满意解. 另外, 通过函数值评价次数可以看出, 利用 BSWA 算法求解背包问题可以大幅降低计算量, 提高求解效率, 这在求解规模较大的背包问题时体现得更加明显.

图 1 和图 2 分别为 BSWA 算法求解背包问题 PB6 和 WEISH20 时适应度函数值的变化情况. 结果表明, BSWA 算法具有较好的离线性能, 能够快速求解到全局最优解并保持良好的种群多样性.

实验 2 文献[8]采用遗传算法(GA)对表 1 中的 9 个背包问题进行了测试, 分别可归属于 3 类, 依次是 Knap15、Knap20、Knap28、Knap39 和 Knap15, SENT1 和 SENT2, WEING7 和 WEING8. 文献[8]中的 GA 算法采用二进制编码(与本文同), 种群规模 $\mu=50$, 变异概率 $p_m = \frac{1}{n}$ (n 为物品数), 交叉概率

$p_c=0.6$, 采用比例选择和单点交叉策略. 另外, 为了防止不可行解, 文献[8]在初始种群的产生中加入了偏倚策略(即根据问题的不同控制个体编码中 0 和 1 的多少), 在适应度函数计算中, 对于非可行解, 按照违反约束的数量进行相应惩罚, 在遗传操作中, 对于最终搜索到的非可行最优解加入贪婪修正策略. 采用 GA 算法对每个背包问题独立运行 100 次, 并对求解到最优解的次数和各个不同的非最优解的次数进行了统计. 用本文 BSWA 算法对以上 9 个背包问题独立运行 100 次, 将 BSWA 算法所得统计结果以及文献[8]中 GA 算法的统计结果列于表 2 中.

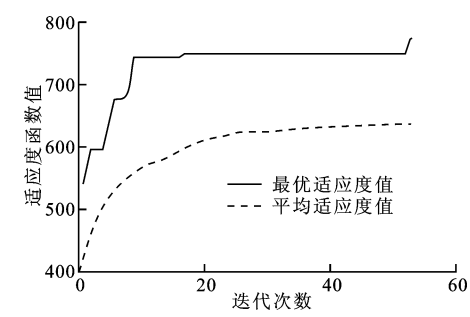


图 1 PB6 的适应度函数值变化情况

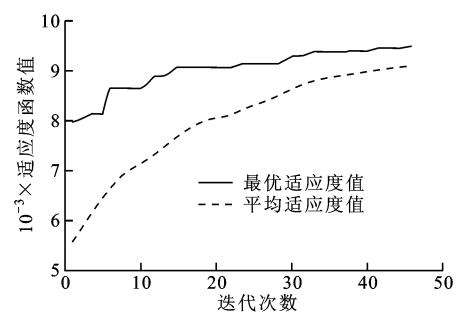


图 2 WEISH20 的适应度函数值变化情况

表 1 55 个标准背包问题的实验结果

问题	n	m	f^*	O_n	f_{mean}	问题	n	m	f^*	O_n	f_{mean}
HP1	28	4	3 418	48	3 417.4	WEISH04	30	5	4 561	50	4 561
HP2	35	4	3 186	8	3 155.3	WEISH05	30	5	4 514	50	4 514
Knap6	6	10	3 800	50	3 800	WEISH06	40	5	5 557	42	5 555
Knap10	10	10	8 706	50	8 706	WEISH07	40	5	5 567	50	5 567
Knap15	15	10	4 015	50	4 015	WEISH08	40	5	5 605	50	5 605
Knap20	20	10	6 120	50	6 120	WEISH09	40	5	5 246	50	5 246
Knap28	28	10	12 400	45	12 399	WEISH10	50	5	6 339	50	6 339
Knap39	39	5	10 618	42	10 615	WEISH11	50	5	5 643	50	5 643
Knap50	50	5	16 537	0	16 465	WEISH12	50	5	6 339	50	6 339
PB1	27	4	3 090	50	3 090	WEISH13	50	5	6 159	50	6 159
PB2	34	4	3 186	6	3 158	WEISH14	60	5	6 954	50	6 954
PB4	29	2	95 168	50	95 168	WEISH15	60	5	7 486	50	7 486
PB5	20	10	2 139	33	2 133	WEISH16	60	5	7 289	50	7 289
PB6	40	30	776	50	776	WEISH17	60	5	8 633	50	8 633
PB7	37	30	1 035	18	1 033	WEISH18	70	5	9 580	31	9 580
SENT1	60	30	7 772	0	6 704	WEISH19	70	5	7 698	50	7 698
SENT2	60	30	8 722	0	8 202	WEISH20	70	5	9 450	50	9 450
WEING1	28	2	141 278	50	141 278	WEISH21	70	5	9 074	50	9 074
WEING2	28	2	130 883	50	130 883	WEISH22	80	5	8 947	50	8 947
WEING3	28	2	95 677	50	95 677	WEISH23	80	5	8 344	33	8 344
WEING4	28	2	119 337	50	119 337	WEISH24	80	5	10 220	50	10 220
WEING5	28	2	98 796	50	98 796	WEISH25	80	5	9 939	50	9 939
WEING6	28	2	130 623	50	130 623	WEISH26	90	5	9 584	50	9 584
WEING7	105	2	1 095 445	49	1 095 443	WEISH27	90	5	9 819	50	9 819
WEING8	105	2	624 319	42	624 197	WEISH28	90	5	9 492	50	9 492
WEISH01	30	5	4 554	50	4 554	WEISH29	90	5	9 410	50	9 410
WEISH02	30	5	4 536	50	4 536	WEISH30	90	5	1 1191	50	11 191
WEISH03	30	5	4 115	50	4 115						

从表 2 可以看出,BSWA 算法求解到最优解的频率明显高于 GA 算法,这充分说明了 BSWA 算法的稳定性和有效性.对于背包问题 Knap50、SENT1 和 SENT2,虽然 GA 算法的求解效果稍微优于

表 2 BSWA 和 GA 对 9 个背包问题的测试结果

项目	n	m	f^*	O_n		f_{mean}	
				OBSWA	GA	BSWA	GA
Knap15	15	10	4 015	100	83	4 015	4 013
Knap20	20	10	6 120	100	33	6 120	6 102
Knap28	28	10	12 400	81	33	12 398	12 375
Knap39	39	5	10 618	80	4	10 594	10 537
Knap50	50	5	16 537	0	1	16 388	16 378
SENT1	60	30	7 772	0	5	6 704	7 626
SENT2	60	30	8 722	0	2	8 203	8 685
WEING7	105	2	1 095 445	98	0	1 095 443	1 093 897
WEING8	105	2	624 319	82	6	624 127	613 383

BSWA 算法,但是,两种算法的求解效果均不理想,这不仅与问题本身的特殊性有关(即全局最优解附近可行解数目稀疏),还与修正策略有关,因为修正策略实施的主要参数为 $\frac{p_i}{a_{ij}}$,这种贪婪修正策略本身不能保证求解到全局最优解. 整体而言,BSWA 算法在求解多维 0-1 背包问题时的性能优于 GA 算法,是一种比较有发展潜力的优化算法.

4 结 论

本文将强调小世界搜索特性的 BSWA 算法用于优化一类典型的组合优化问题——多维背包问题. 相关实验结果及与 GA 算法的对比分析均表明,BSWA 算法求解背包问题时速度快,稳定性好,在保持种群多样性和求解精度方面同样具有良好的性能,体现出解决组合优化问题的极大潜力. 对于部分 BSWA 算法不能搜索到最优解的背包问题,尝试引入罚函数方法(即对不可行解进行保留),将罚函数法和贪婪修正策略结合将是一个值得探索的方法. 另外,将 BSWA 算法用于其他组合优化问题中,例如 TSP 问题、通信网络设计和分布式系统优化等,并针对不同问题的特点对 BSWA 算法进行改进研究是未来研究的一个重点.

参考文献:

[1] 玄光男,程润伟. 遗传算法与工程优化 [M]. 于歆杰,周根贵,译. 北京:清华大学出版社,2004:55-60.

[2] MARTELLO S, PISINGER D, TOTH P. New trends in exact algorithms for the 0-1 knapsack problem[J]. European Journal of Operational Research, 2000, 123 (2):325-332.

[3] AKCAY Y, LI Haijun, XU S H. Greedy algorithm for the general multidimensional knapsack problem [J]. Annals of Operations Research, 2007, 150(1):17-29.

[4] CHU P C, BEASLEY J E. A genetic algorithm for the multidimensional knapsack problem [J]. Journal of Heuristics, 1998, 4(1):63-86.

[5] 杜海峰, 庄健, 张进华, 等. 用于函数优化的小世界优化算法[J]. 西安交通大学学报, 2005, 39(9):1011-1015.

DU Haifeng, ZHUANG Jian, ZHANG Jinhua, et al. Small-world phenomenon for function optimization [J]. Journal of Xi'an Jiaotong University, 2005, 39 (9):1011-1015.

[6] MICHALEWICZ Z. Genetic algorithms+data structures = evolution programs [M]. 2nd ed. Berlin, Germany: Spring-Verlag, 1994.

[7] BEASLEY J E. OR-library: distribution test problems by electronic mail [J]. Journal of Operational Research Society, 1990, 41(11): 1069-1072.

[8] KHURI S, BÄCK T, HEITKÖTTER J. The zero/one multiple knapsack problem and genetic algorithms [C]// Proceedings of the 1994 ACM Symposium on Applied Computing. New York, USA: ACM Press, 1994: 88-193.

(编辑 刘杨)