

文章编号: 1001-0920(2009)04-0617-04

基于概率突跳和模拟退火的改进自适应微粒群算法

谷良贤, 王轶鹏, 龚春林

(西北工业大学 航天学院, 西安 710072)

摘要: 在两种微粒群算法分析的基础上, 针对算法存在局部最优和后期振荡的现象, 提出一种改进自适应微粒群算法. 新算法引入概率突跳因子改变了原算法中微粒的速度更新公式, 引入模拟退火接受准则抑制了概率突跳的不可控制性. 典型函数寻优结果表明, 新算法能很快地收敛到全局最优解, 大幅度降低了达到最优值所需要的迭代数, 同时提高了算法的收敛率和收敛精度, 在跳出局部搜索的能力上远优于标准微粒群算法和自适应微粒群算法, 稳定性好.

关键词: 微粒群优化; 模拟退火; 自适应; 概率突跳

中图分类号: TP18; TP301.6 **文献标识码:** A

Modified adaptive particle swarm optimization algorithm based on probabilistic leap and simulated annealing

GU Liang-xian, WANG Yi-peng, GONG Chun-lin

(College of Astronautics, Northwestern Polytechnical University, Xi'an 710072, China. Correspondent: WANG Yi-peng, E-mail: kaoyan200606@163.com)

Abstract: On the basis of analyzing two particle swarm optimization (PSO) algorithms, the standard PSO (SPSO) and self-adapting PSO (SAPSO), a modified adapting PSO (MAPSO) algorithm is proposed to solve the problem that PSO may trap to local optimum and fluctuation during later period. In this algorithm, the probabilistic leap factor is introduced to modify the velocity updating and the acceptable rule of simulated annealing is applied to restrain the uncontrollability of probabilistic leap. The results of typical optimization show that this algorithm has better accuracy and convergence rate as well as fewer iteration numbers in approaching the global optimization than SPSO and SAPSO algorithms. This algorithm is also superior to SPSO and SAPSO algorithms in stability and ability of breaking off local search.

Key words: PSO; Simulated annealing; Self-adapting; Probabilistic leap

1 引言

微粒群算法 (PSO) 是一种启发式全局进化寻优算法, 其思想来源于对鸟群捕食行为的研究, 由 Eberhart 等^[1]首次提出. 算法基于群体迭代, 通过微粒个体之间的协作在解空间中追随最优微粒进行搜索寻优. 同遗传算法和蚁群算法相比, PSO 具有算法简单、可调参数少、容易实现等优点, 目前已成功应用于结构设计、神经网络、电磁场、多目标规划和任务调度等工程优化问题中.

PSO 算法收敛速度较快, 但会出现早熟收敛、甚至不收敛的情况, 尤其对于多峰值函数而言不能令人满意^[2]. 对 PSO 算法进行改进从而进一步提高

算法的性能成为该领域一个研究热点, 相继出现了许多改进优化算法: Shi 等^[3]提出了引入惯性权重的标准微粒群算法; Natsuki 等^[4]给出具有高斯变异的微粒群算法; Mahfour^[5]提出了带收缩因子的自适应微粒群算法; 刘建华等^[6]提出了基于相似度的微粒群算法等. 文献[7]引入概率突跳因子之后微粒仍然存在较大的不可控制性, 文献[8]通过接受准则的引入, 虽然在一定程度上提高了算法的稳定性, 但寻优能力却以牺牲时间为代价.

本文基于自适应微粒群算法, 针对以上问题提出改进措施, 对 4 个典型优化函数进行测试, 与 SPSO, SAPSO 算法测试数据结果进行对比分析. 仿

收稿日期: 2008-03-18; 修回日期: 2008-06-24.

基金项目: 国家自然科学基金项目 (60774087); 航天支撑技术基金项目 (0711205).

作者简介: 谷良贤 (1957—), 女, 西安人, 教授, 博士生导师, 从事飞行器总体设计、多学科优化设计等研究; 王轶鹏 (1982—), 男, 西安人, 助教, 硕士生, 从事学科设计优化、飞行器总体设计的研究.

真结果表明,新算法进一步提升了 PSO 算法的性能和计算精度.

2 PSO 算法数学模型及主要参数分析

在一个 D 维搜索空间 $s \in R^D$ 和 m 个微粒组成的种群中,记 $x_i = (x_{i1}, x_{i2}, \dots, x_{iD})$, $v_i = (v_{i1}, v_{i2}, \dots, v_{iD})$, $p_i = (p_{i1}, p_{i2}, \dots, p_{iD})$, $p_g = (x_{g1}, x_{g2}, \dots, x_{gD})$ 分别为微粒 i 的当前位置、当前飞行速度、个体历史最优位置和种群历史最优位置. 当前,对 PSO 算法的改进是采用对微粒位置 X 和飞行速度 V 的改变来实现的,下面给出微粒位置 X 和飞行速度 V 的两种典型更新规则.

2.1 标准微粒群算法 (SPSO)

标准微粒群算法是在基本 PSO 算法基础上,通过引入惯性权重 来控制历史速度对下一时刻速度的影响,其微粒位置和速度更新关系式^[3]为

$$\begin{cases} v_{id}^{t+1} = v_{id}^t + c_1 r_1 (p_{id}^t - x_{id}^t) + \\ \quad c_2 r_2 (p_{gd}^t - x_{id}^t), \\ x_{id}^{t+1} = x_{id}^t + v_{id}^{t+1}. \end{cases} \quad (1)$$

其中 c_1 和 c_2 为学习因子,又称加速常数,为非负数,通常取 $c_1 = c_2 = 2$; r_1 和 r_2 为随机因子, $r_1, r_2 \in U[0, 1]$; w 为惯性权重; t 为当前迭代数,又称进化代数,记最大迭代数为 t_{\max} ,则 $t \in [0, t_{\max}]$.

当 v_{\max} 增加时,可通过减小 w 来平衡搜索. 为加强 PSO 的全局搜索能力,迭代初期 w 取值较大,以利于跳出局部极小点;迭代后期 w 取值相应减小,以加强算法的局部搜索能力,使得算法在最优解周围精细搜索,获得全局最优解. 总之, w 的引入使得所需的迭代次数减小^[9]. 标准 PSO 算法是目前使用最广泛的 PSO 算法.

2.2 自适应微粒群算法 (SAPSO)

为了改进 SPSO 算法的空间搜索效率, Mahfour 于 2004 年提出了自适应算法,其位置更新模式与 SPSO 相同,速度更新模型^[5]为

$$\begin{cases} v_{id}^{t+1} = v_{id}^t + a [c_1 r_1 (p_{id}^t - x_{id}^t) + \\ \quad c_2 r_2 (p_{gd}^t - x_{id}^t)], \\ a = a_0 + \frac{t}{t_{\max}}. \end{cases} \quad (2)$$

从 SAPSO 算法中分析可知:随着迭代次数的增加, a 值逐渐减小,PSO 的全局搜索能力逐渐减弱,相应的局部搜索能力逐渐增强. 为此, SAPSO 算法中引入加速项 a ,随着迭代次数的增加, a 值越来越大,提高了算法的全局搜索能力,从而降低 PSO 陷入局部最优的概率. a 的引入相当于抑制了 w 的递减速度,在一定程度上调和了算法的全局搜索能力与局部搜索能力之间的矛盾.

3 改进自适应微粒群算法 (MAPSO)

本文基于自适应微粒群算法,提出一种改进的微粒群优化算法,位置更新模式仍然与 SPSO 相同.

3.1 算法改进机理分析

(1) 引入概率突跳因子

SPSO 优化算法中,微粒位置的更新是行为惯性、自我认知和社会性 3 个方面共同作用的结果. 社会系统中的个体根据自己的行为惯性、认知能力和社会环境所做出的判断,受主客观因素和认知水平的影 响,总存在对形势的误判、决策的失误与盲从等现象^[7]. 为减轻由于盲从和误判所引起的误差,本文通过引入突跳因子来改变微粒的既定速度,使得微粒以一定概率跳离原算法得到的位置. 基于 SAPSO 优化算法,新的速度更新公式变为

$$\begin{cases} v_{id}^{t+1} = 0, & r_3 > c_3, \dots; \\ v_{id}^{t+1} = v_{id}^t + a [c_1 r_1 (p_{id}^t - x_{id}^t) + \\ \quad c_2 r_2 (p_{gd}^t - x_{id}^t)], & \text{otherwise.} \end{cases} \quad (3)$$

其中 $r_3 \in U[0, 1]$ 和 $c_3 \in [0, 1]$ 是预先指定的常数. 当 $c_3 = 0$ 时,所有的微粒都将停滞在原来的位置保持不动,当 $c_3 = 1$ 时,新算法就退化为自适应粒子群算法. 显然, c_3 的引入改变了 SAPSO 算法中微粒飞行的方向和距离,使得微粒产生了突跳行为,突跳概率为 $1 - c_3$.

(2) 引入模拟退火思想的接收准则

模拟退火算法是 20 世纪 80 年代发展起来的一种用于求解大规模组合优化问题的随机算法. 它源于对固体退火过程的模拟,采用 Metropolis 接受准则,并用一组称为冷却进度表的参数控制算法进程,使算法能在多项式时间里给出一个近似最优解. 作为对局部搜索算法的拓展,模拟退火算法的基本思想是从给定解开始,从某局部邻域随机产生一个新解,在 Metropolis 接受准则指导下以一定概率接受新解. 受该算法接受准则的启发,本文以适应度函数变化量 F 类比模拟退火算法中的能量差 E ,为了简化算法结构,设目标函数允许变坏范围 $(- \delta, \delta)$. 其中: δ 为目标变坏的上限, $-\delta$ 为目标变坏的下限. 只要 $|F| < \delta$,则更新微粒位置,否则微粒位置保持不变. 文中适应度函数变化量取为理论最优值与算法寻优过程中所找到的实际最优值之间的差值. 为了提高算法的收敛精度,设 $\delta > 0$,规定 $\delta \in U(0; \delta)$,即 $\delta \in (-\delta, \delta)$,的具体取值应该根据不同的问题所要求的精度而变化.

(3) 指数递减策略代替线性递减策略

在 MAPSO 模型中,对惯性权值的选择,本文抛弃“线性”递减策略

$$= (0 - f) \left(1 - \frac{t}{t_{\max}}\right),$$

采用指数递减策略

$$= f \left(\frac{0}{f}\right)^{\frac{1}{1+10\frac{t}{t_{\max}}}}$$

采用指数递减策略,使得算法在迭代初期具有更大的突跳性,后期平缓过渡更有利于算法的稳定。

3.2 MAPSO 算法的步骤及流程

MAPSO 算法的步骤如下:

Step1: 在搜索空间内随机初始化微粒群的位置、速度、个体历史最优值和种群历史最优值;

Step2: 比较微粒当前位置和历史最优位置,更新 p_i 和 p_g ;

Step3: 计算 r_3 , 判断 r_3 和 c_3 的大小,执行一次突跳机制,完成速度和位置更新;

Step4: 执行模拟退火接受准则,如果 $|F| > e^{-\frac{1}{F}}$, 则令 $v_{id}^{t+1} = 0$, 进行一次位置更新;否则,直接进行一次位置更新,完成微粒当前速度和位置的更新;

Step5: 判断是否满足终止条件,如满足则退出,否则转到 Step1.

MAPSO 算法的流程如图 1 所示。

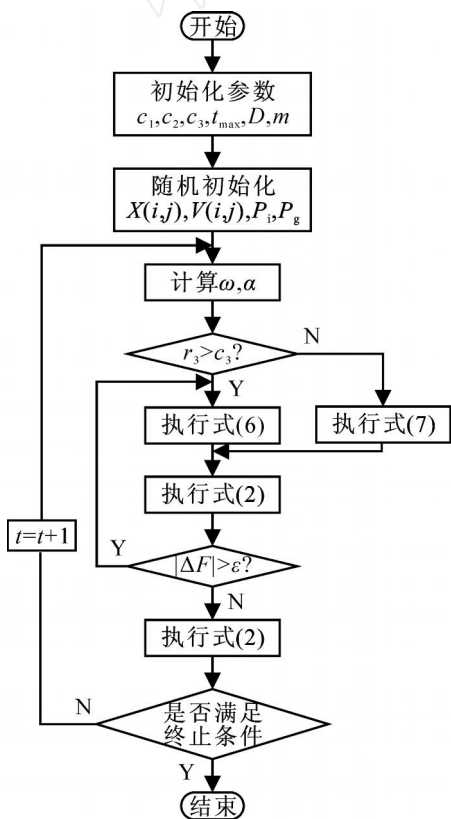


图 1 MAPSO 算法流程图

4 算法验证

4.1 测试函数

为检验本文提出的算法的有效性,选择以下 4 个典型函数进行测试:

1) Sphere 函数

$$f_1(X) = \sum_{i=1}^n x_i^2, x_i \in [-100, 100],$$

$\min f_1(0, \dots, 0) = 0$ 为单峰值函数

2) Griewangk's 函数

$$f_2(X) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1,$$

$x_i \in [-600, 600], \min f_2(0, \dots, 0) = 0.$

3) Schaffer 函数

$$f_3(X) = \frac{\sin^2 \sqrt{0.5(x_1^2 + x_2^2)} - 0.5}{[1 + 0.001(x_1^2 + x_2^2)]^2} - 0.5,$$

$x_i \in [-100, 100], \min f_3(0, 0) = -1,$

在距离最优解大约 3.14 的半径范围内存在无限多次振荡。

4) Shubert 函数

$$f_4(X) = \sum_{j=1}^5 j \cos[(j+1)x_1 + j] \times \sum_{j=1}^5 j \cos[(j+1)x_2 + j],$$

$x_i \in [-100, 100]$ 有 760 个局部最小值,其中的 18 个是全局最小,其值约为 -186.7309.

4.2 仿真结果

对于算法 SPSO, SAPSO 和 MAPSO, 取 $t_{\max} = 1000, c_1 = c_2 = 2.0, \text{种群规模 } m = 25, \omega = 0.95, \varphi = 0.4, a_0 = 0.6,$ 微粒位置和速度的边界值等于相应的自变量的边界值。

算法的有效性和稳定性能采用达优率、平均迭代次数和收敛精度(或相对误差)3 个方面来衡量,这里的收敛率定义为在最大迭代次数内算法找到最优值的次数与总的运行次数的比值.平均迭代数表示收敛情况下的平均迭代数,对于平均迭代数为小数的情况统一采用下取整运算。

算法收敛的终止规则为:

1) 引入接受准则后允许目标变坏的上限 $\Delta F = e^{-10}$, 同时最小误差限取值等同于 ΔF (表示相邻两代最优值之间的误差),如果相邻最优值误差的绝对值不大于这个最小误差,则认为算法已经找到最优值。

2) t_{\max} 为最大迭代数,如果算法的迭代次数超过该值都没有能够找到满足指定精度的最优值,则认为寻优失败。

另外由于微粒群算法的随机性,只有进行大量测试才能评估算法的性能,所以各算法对各函数分别独立运行 50 次.仿真结果如图 2 ~ 图 5 和表 1 所示。

从图 2 ~ 图 5 和表 1 可以看出,对于所有测试函数而言,新算法的优化效率都得到了大幅度的提

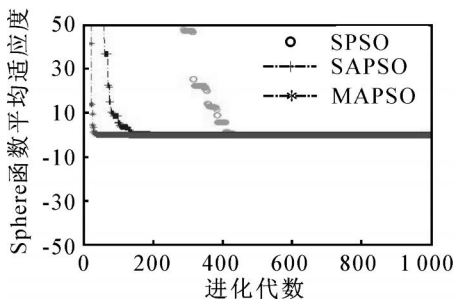


图2 Sphere 函数平均适应度变化曲线

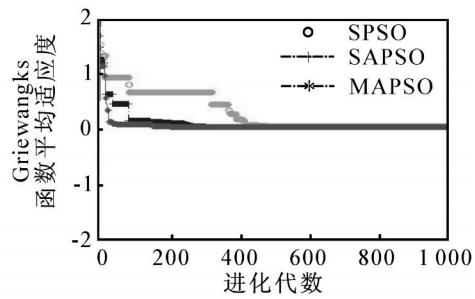


图3 Griewangk s 函数平均适应度变化曲线

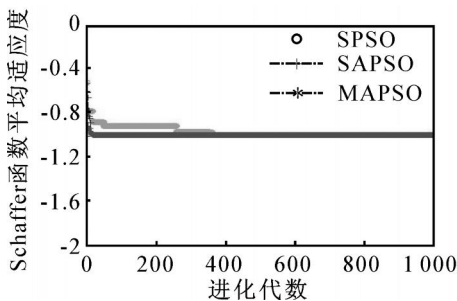


图4 Schaffer 函数平均适应度变化曲线

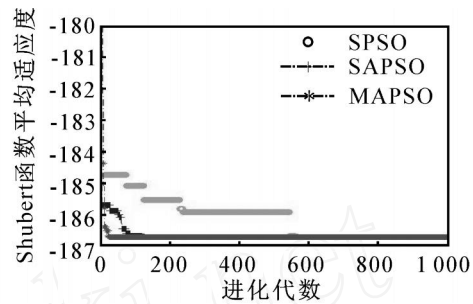


图5 Shubert 函数平均适应度变化曲线

表1 测试仿真结果及性能分析

函数	适应度均值			平均迭代次数			平均收敛精度(相对误差)			收敛率 / %		
	SPSO	SAPSO	MAPSO	SPSO	SAPSO	MAPSO	SPSO	SAPSO	MAPSO	SPSO	SAPSO	MAPSO
Sphere	1.0E-44	4.2E-13	1.7E-57	626	470	97	E-44	E-13	E-57	100	100	100
Griewangk	2.3E-18	0	0	546	353	64	E-18	0	0	98	100	100
Schaffer	-0.99436	-0.99728	-0.99797	521	253	40	5.6E-03	2.7E-03	2.0E-03	48	72	78
Shubert	-1.866526 E+02	-1.867307 E+02	-1.867309 E+02	623	335	106	4.2E-04	1.1E-06	0	72	98	100

高. 对于单峰值函数 f_1 , SPSO 在计算精度上明显优于 SAPSO, 较 MAPSO 算法在计算精度上低 13 个百分点; 对于函数 f_2 , SPSO 在计算精度上的优势明显不如 SAPSO 和 MAPSO, 后两种算法达到了零误差; 对于复杂多峰值函数 f_4 , MAPSO 算法在计算精度和跳出局部搜索能力上的优势明显地体现了出来, 达到了零误差, 收敛率达到了百分之百; 对于函数 f_3 , 3 种算法的测试结果较之其他 3 个测试函数在计算精度和收敛率方面都有所下降, 但 MAPSO 算法较 SPSO, SAPSO 的优势仍然可以看到.

5 结 论

在自适应微粒群优化算法数学模型的基础上, 对原算法进行了改进. 从社会心理学角度引入概率突跳因子, 受模拟退火思想启发引入接受准则, 并采用“指数”型递减策略改变原微粒速度更新公式, 使得微粒飞行时以一定概率在解空间内发生连续性突跳, 并使得微粒各维元素的改变规则不尽相同, 大大丰富了微粒种群的多样性, 增强了寻优过程中的稳定性. 通过对 4 个典型优化函数的数值测试结果表明, 本文提出的 MAPSO 算法, 在收敛速度、达优率

和收敛精度上都优于 SPSO 和 SAPSO 算法, 稳定性好.

参考文献(References)

- [1] Kennedy J, Eberhart R. Particle swarm optimization [C]. IEEE Int Conf on Neural Networks. Piscataway, 1995: 1942-1948.
- [2] Eberhart R C, Shi Y H. Comparing inertia weights and constriction factors in particle swarm optimization [C]. Proc 2000 Congress Evolutionary Computation. Piscataway: IEEE Press, 2000: 84-88.
- [3] Shi Y, Eberhart R C. A modified particle swarm optimizer [C]. Proc of the IEEE Int Conf on Evolutionary Computation. Piscataway: IEEE Press, 1998: 69-73.
- [4] Natsuki Higashi, Hitoshi Iba. Particle swarm optimization with Gaussian mutation [C]. Proc of the Congress on Evolutionary Computation. Washington DC, 2003: 72-79.
- [5] Mahfouf M, Chen M Y, Linkens D A. Adaptive weighted swarm optimization for multi-objective optimal design of alloy steels [J]. Lecture Notes in Computer Science, 2004, 3242: 762-771. (下转第 627 页)

$$\gamma(t) = \sqrt{\int_0^t z^T(\tau) z(\tau) d\tau / \int_0^t w^T(\tau) w(\tau) d\tau}$$

从图 3 可以看出,在正常情况下 $\gamma(t)$ 迅速小于 0.5619,并且在执行器故障情况下也远小于 0.8.

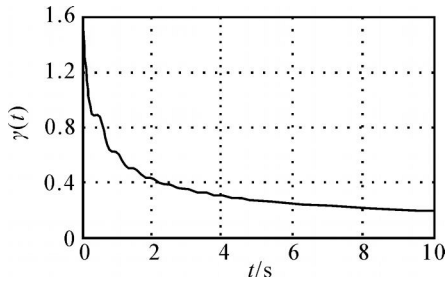


图 3 被控输出能量与外扰能量比的平方根

5 结 论

本文给出了两种基于观测器的 H_∞ 模糊可靠控制设计方法:两步法和相似变换法.两种方法均能将 BMI 问题转化为 LMI 问题.本文所提出的相似变换法不需要满足文献[3]方法所要求的局部输入矩阵列满秩的条件.分析了两种方法在适用范围和保守性方面的不同之处,采用何种设计方法需根据实际情况具体选择.仿真示例验证了本文方法的有效性.

参考文献(References)

- [1] Takagi T, Sugeno M. Fuzzy identification of systems and its applications to modeling and control[J]. IEEE Trans on Systems, Man and Cybernetics, 1985, 15(1): 116-132.
- [2] Wu H N, Li H X. H_∞ fuzzy observer-based control for a class of nonlinear distributed parameter systems with control constraints[J]. IEEE Trans on Fuzzy Systems, 2008, 16(2): 506-516.
- [3] Lo J C, Lin M L. Existence of similarity transformation converting BMIs to LMIs[J]. IEEE Trans on Fuzzy Systems, 2007, 15(5): 840-851.
- [4] Wu H N, Zhang H Y. Reliable mixed L_2/H_∞ fuzzy static output feedback control for nonlinear systems with sensor faults[J]. Automatica, 2005, 41(11): 1925-1932.
- [5] Wu H N, Zhang H Y. Reliable H_∞ fuzzy control for continuous-time nonlinear systems with actuator failures[J]. IEEE Trans on Fuzzy Systems, 2006, 14(5): 609-618.
- [6] Chen B, Liu X P. Reliable control design of fuzzy dynamic systems with time-varying delay[J]. Fuzzy Sets and Systems, 2004, 146(3): 349-374.
- [7] Assawinchaichote W, Nguang S K, Shi P, et al. H_∞ fuzzy state-feedback control design for nonlinear systems with D -stability constraints: An LMI approach[J]. Mathematics and Computers in Simulation, 2008, 78(4): 514-531.
- [8] Yang G H, Wang J L, Soh Y C. Reliable H_∞ controller design for linear systems[J]. Automatica, 2001, 37(5): 717-725.
- [9] He Y, Wu M, Jin H S. Improved bounded-real lemma representation and H_∞ control of systems with polytopic uncertainties[J]. IEEE Trans on circuits and systems, 2005, 52(7): 380-383.
- [10] Tuan H D, Apkarian P, Narikiyo T, et al. Parameterized linear matrix inequality technique in fuzzy control systems design[J]. IEEE Trans on Fuzzy Systems, 2001, 9(2): 324-332.

(上接第 620 页)

- [6] 刘建华,樊晓平,瞿志华.一种基于相似度的新型粒子群算法[J].控制与决策,2007,22(10):1155-1159.
(Liu J H, Fan X P, Qu Z H. A new particle swarm optimization algorithm based on similarity[J]. Control and Decision, 2007, 22(10): 1155-1159.)
- [7] 姜海明,谢康,王亚非.按概率突跳的改进微粒群优化算法[J].吉林大学学报,2007,37(1):141-145.
(Jiang H M, Xie K, Wang Y F. Modified particle swarm optimization via probabilistic leap[J]. J of Jilin University, 2007, 37(1): 141-145.)
- [8] 高尚,杨静宇,吴小俊,等.基于模拟退火算法思想的粒子群优化算法[J].计算机应用与软件,2005,22(2):103-104.
(Gao S, Yang J Y, Wu X J, et al. Particle swarm optimization based on the ideal of simulated annealing algorithm[J]. Computer Applications and Software, 2005, 22(2): 103-104.)
- [9] Shi Y, Eberhart R C. Parameter selection in particle swarm optimization[C]. Annual Conf on Evolutionary Programming. San Diego, 1998: 591-600.