

继承优秀染色体片段的 PSO 算法求解 TSP 问题

程 乐, 张洪斌

(淮安信息职业技术学院 计算机科学与工程系, 江苏 淮安 223003)

摘 要: 粒子群优化算法(PSO)提出至今一直未能有效解决离散及组合优化问题, TSP 问题是组合优化问题中一个典型的 NP 问题. 文中参考了离散粒子群算法(DPSO)和遗传算法(GA)解决 TSP 问题的成功经验, 提出了一种继承优秀染色体片段的 PSO 算法(ECFG-PSO). 为避免早熟, 在算法中加入了局部查找和二次初始化策略. 实验证明 ECFG-PSO 算法解决 TSP 问题的效率和规模优于 DPSO 算法.

关键词: 粒子群优化算法; TSP; DPSO; 遗传算法; ECFG-PSO

中图分类号: TP18

文献标识码: A

文章编号: 1000-7180(2010)07-0202-04

Excellent Chromosome Fragments Genetic PSO for Solving Traveling Salesman Problem

CHENG Le, ZHANG Hong-bin

(Department of Computer Science and Engineering, Huaian College of Information Technology, Huaian 223003, China)

Abstract: Particle swarm optimization (PSO) has been applied to many practical continuous optimization problems. However, it could not be extended to solve discrete and combinatorial optimization problems effectively. Traveling salesman problem (TSP) is a typical NP-hard problem. In view of the success of and GA in discrete and continuous optimization problems, this paper proposes a new Excellent Chromosome Fragments Genetic PSO (ECFG-PSO) for TSP. To prevent premature convergence of ECFG-PSO, local searching and the secondary initialization were added to the algorithm. The experiments show that the ECFG-PSO is superior to the DPSO existed in time efficiency and problem size.

Key words: PSO; TSP; DPSO; GA; ECFG-PSO

1 引言

粒子群优化算法(Particle Swarm Optimization, PSO)于 1995 年提出. 由于 PSO 算法既保持了传统进化算法的深刻群体智慧背景, 同时又有自己许多良好的优化性能. 因此, PSO 一经提出立刻引起了进化计算领域学者们的广泛关注^[1-3]. 旅行商问题(Travel Salesman Problem, TSP)是一个典型的 NP 完全问题^[3]. 虽然其数学描述非常简单, 但却无法找到一个确定的算法可以在多项式时间内求解 TSP 问题. 近年来仿生学优化理论的发展, 提出了采用模拟自然进化过程求解 TSP 问题的一系列方

法, 研究成果多集中在遗传算法(Genetic Algorithm, GA)^[4]、蚁群算法(Ant Colony Optimization, ACO)^[5].

PSO 算法在连续空间优化问题上已经取得了很好的效果^[1-2], 但是将 PSO 应用于 TSP 这样的离散组合优化问题还是一种全新的挑战. 为了使 PSO 应用于各类离散问题, Kennedy 和 Eberhart 提出二进制 PSO 算法^[6], 通过优化可连续变化的二进制概率(即二进制变量为 1 的概率)达到间接优化二进制变量的目的. 但是该间接优化策略是根据概率而非算法本身确定二进制变量, 因此未能充分利用粒子群优化算法的性能. 文献[7]给出了一种利用整体

置换(Permutation)求解 TSP 问题的方法.该方法主要通过比较粒子与全局最优(gbest)和本地最优(lbest)的差别定义交换序列,将交换序列作用于粒子直接得到新的粒子进行进化.文献[7]证明了粒子群算法可以解决 TSP 问题,但也指出此方法性能有所欠缺.

GA 算法通过对当前解交叉、变异实现遗传.通过优胜劣汰保持解规模不变.TSP 的解是一个不重复的节点序列,同时也是一组编码组合.因此 TSP 问题比较合适用 GA 算法求解.文献[8]对 GA 在求解过程中基因片段保存机制进行了深入研究,并且取得了很好的效果.

文中参考了遗传算法中染色体的构造方法和遗传机制,在 DPSO 算法的基础上,提出了继承优秀染色体片段的 PSO(Excellent Chromosome Fragments Genetic PSO, ECFG-PSO)算法,并在该算法中加入了局部搜索和二次初始化机制.实验证明 ECFG-PSO 算法收敛性强,计算代价低.

2 TSP 模型和基本的 PSO 算法

TSP 问题数学描述如下^[2]: 设 $C = \{c_1, c_2, \dots, c_n\}$ 是一个城市的集合;而 $Z = \{b_{ij} | c_i, c_j \in C\}$ 表示 C 中两两连接的边;TSP 问题目的就是寻找有向图 $G = (C, Z)$ 的一个长度最短 Hamilton 圈.

PSO 算法初始化为一群随机粒子(随机解).然后通过迭代找到最优解.在每一次迭代中,粒子通过跟踪两个“极值”来更新自己.一个是粒子本身所找到的最优解(p_{id}).另一个是整个种群目前找到的最优解 p_{gd} .找到这两个最优解后,粒子根据如下公式来更新自己的位置:

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

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

ω 表示惯性权值, v_{id} 表示第 i 个粒子在 d 维空间的速度, x_{id} 表示第 i 个粒子在 d 维空间的坐标. p_{id} 和 p_{gd} 定义如前, $\text{rand}()$ 是介于 $[0, 1]$ 之间的随机数, c_1, c_2 是学习因子,通常 $c_1 = c_2 = 2$.

式(1)的“ $\omega \cdot v_{id}$ ”部分由微粒上一次速度的惯性引起,为“性质”部分;“ $c_1 \cdot \text{rand}() \cdot (p_{id} - x_{id})$ ”部分为“认知”部分,表示微粒本身的思考,即微粒本身的信息对自己下一步行为的影响;“ $c_2 \cdot \text{rand}() \cdot (p_{gd} - x_{id})$ ”部分为“社会”部分,表示微粒间的信息共享和相互合作,即群体信息对微粒下一步行为

的影响.

3 ECFG-PSO 算法

3.1 应用于 TSP 问题的改进的 PSO 算法

设问题规模为 n , 即 TSP 有 n 个节点, 则一个解序列 S 可以表示为

$$S = (a_i), i = 1, \dots, n$$

为了将 PSO 算法应用于 TSP 问题, 给出如下定义:

定义 1 交换算子(Swap Operator, SO). $SO(i_1, i_2)$ 为节点 a_{i_1} 和 a_{i_2} 交换位置的行为. 那么一个解 S 交换两个节点的行为可以表示为 $S = S + SO(i_1, i_2)$.

定义 2 交换序列(Swap Sequence, SS). $SS = (SO_1 + SO_2 + \dots, SO_n)$ 为一系列交换节点位置的行为. 那么一个解 S 交换内部节点位置的一系列行为可以表示为 $S = S + SS$.

定义 3 基本交换序列(Basic Swap Sequence, BSS). BSS 为从某一初始解出发达到某一个目标解所使用的 SO 个数最少的 SS.

定义 4 运算符“+”. 用于连接一个解和一个算子, 表示这个算子作用于这个解.

定义 5 运算符“-”. 用于连接两个解, 生成一个 SS. 从一个解 B 经过一个 SS 达另一个解序列 A 可以表示为: $SS = A - B$.

定义 6 运算符“ $\oplus$ ”. 用于连接两个 SS, 生成新的 SS 可以表示为 $SS' = SS_1 \oplus SS_2$.

结合以上六个定义, 给出下面 BSS 的求法.

例 1 假设问题规模为 5 的 TSP 中有解序列 A 和 B , 下面给出从 B 到 A 的 BSS 求解过程, 即 $A - B$ 的求解过程如下:

$$A: (1, 2, 3, 4, 5)$$

$$B: (2, 3, 1, 5, 4)$$

Step1: $A(1) = B(3) = 1$, 所以第一个 SO 为 $SO(1, 3)$, 即 $B_1 = B + SO(1, 3)$.

$$B_1 = (1, 3, 2, 4, 5).$$

Step2: $A(2) = B_1(3) = 2$, 所以第二个 SO 为 $SO(2, 3)$, 即 $B_2 = B_1 + SO(2, 3)$.

$$B_2 = (1, 2, 3, 5, 4).$$

Step3: $A(4) = B_2(5) = 4$, 所以第三个 SO 为 $SO(4, 5)$, 即 $B_3 = B_2 + SO(4, 5)$.

$$B_3 = (1, 2, 3, 4, 5) = A.$$

结束.

例 1 中得到 $BSS = (SO(1, 3), SO(2, 3),$

$SO(4,5)$). 根据以上定义,改进后的 PSO 公式为:

$$v_{id} = v_{id} \oplus \text{rand}() \cdot (p_{id} - x_{id}) \oplus \text{rand}() \cdot (p_{gd} - x_{id}) \quad (3)$$

$$x_{id} = x_{id} \oplus v_{id} \quad (4)$$

式(3)中 $\text{rand}()$ 是介于 $[0,1]$ 之间的随机数,其限制了由 $(p_{id} - x_{id})$ 和 $(p_{gd} - x_{id})$ 计算得出的 BSS 中发挥作用的 SO 的数量.

3.2 ECFG-PSO 算法

文中参考了此类 GA 算法应用于 TSP 问题的成功经验,运用 PSO 算法求解 TSP 问题的同时把每一个 $S = (a_i) (i = 1 \cdots n)$ 看做一条染色体.在规模较大的 TSP 问题中,SS 对一个 S 进行一系列交换节点的操作时得到 S 未必是最优解,但是 S 中的某个节点片段 P 可能是合理的优秀基因片段.那么由 S 经过 SS 变换生成下一个解序列 S' 中 P 片段中的节点顺序和位置保持不受干扰会对最终找到最优解有所帮助.为了明确优秀染色体片段 (Excellent Chromosome Fragments, ECF) 的含义以及优秀染色体片段的查找方法,给出如下定义:

定义 7 优秀染色体片段长度 λ . 表示优秀染色体片段包含的节点个数: $\lambda \in (1, n)$.

定义 8 优秀染色体符合度 γ . 如果包含节点片段 P 的粒子个数为 m , 粒子群规模为 M . 如果 $m/M \geq \gamma$ 则称 P 为优秀染色体片段 (ECF), $\gamma \in (0, 1)$.

如第 2 节所述,在 PSO 中整个群拥有一个全局最优解 P_{gd} , 每个粒子对应着一个自己目前为止找到的最优解 P_{id} . 假设 TSP 问题中有 n 个城市, 粒子群规模为 M , 查找 ECF 之前需要预先定义优秀染色体片段长度 λ 和优秀染色体符合度 γ . 生成 ECF 的策略是首先在 P_{gd} 随机查找出一段长度为 λ 的连续解片段 P, 然后在所有的 P_{id} 中查找含有 P 的 P_{id} 并记录其个数 m , 当 $m/M \geq \gamma$ 时, 称 P 为 ECF. 这样粒子在飞行时由上一次迭代生成的 ECF, 将被继承. 即在本次迭代执行公式(4)时所有含有 ECF 粒子的 ECF 片段将不受干扰.

例 2 以 3.1 节中的解 A 和 B 为例, 说明优秀染色体片段作用于求解 BSS 的过程如下:

$$A = (1, 2, 3, 4, 5)$$

$$B = (2, < 3, 1 >, 5, 4)$$

这里假设 B 中的 3、1 节点组成的序列片段为 ECF, 执行 $SS = A - B$.

Step1: $A(1) = B(3) = 1$, 因为 B 中 1 在 ECF 中, 所以无 SO.

$$B_1 = (2, < 3, 1 >, 5, 4)$$

Step2: $A(2) = B_1(1) = 2$, B 中下标 2 涉及到 ECF, ECF 不受干扰, 因此无 SO.

$$B_2 = (2, < 3, 1 >, 5, 4)$$

Step3: $A(3) = B_2(2) = 3$; 因为 B 中 1 在 ECF 中, 所以无 SO.

$$B_3 = (2, < 3, 1 >, 5, 4)$$

Step4: $A(4) = B_3(5) = 4$, 所以 $B_4 = B_3 + SO(4, 5)$

$$B_4 = (2, < 3, 1 >, 4, 5)$$

结束.

例 2 中得到 $SS = (SO(4, 5))$ 为所求的 BSS. 相对于例 1, 例 2 在保证优秀染色体片段不受干扰的情况下, 得出 SS 中的 SO 较少. 意味着算法的效率会提高. 而且得出的新解更具合理性.

3.3 局部查找策略与二次初始化

在 ECFG-PSO 算法迭代初期, 在一次迭代中, 对每个粒子算法的执行策略是执行一次公式(3) 得出一个 BSS, 然后 BSS 作用于粒子生成新的 TSP 解, 根据每个粒子的在本次迭代中的解, 更新 p_{id} 和 p_{gd} , 为下一次迭代做准备. 策略中 BSS 的使用可以使算法迅速收敛.

在算法迭代末期, 算法所得当前解已经相对较优的情况下, 每一次交换 TSP 中两个城市节点的位置, 即一个执行 SO 可能对最终查找到最优解有很大的影响. 因此 ECFG-PSO 算法迭代末期, 算法在执行每个 SO 后都将判断其适应值, 更新 p_{id} 和 p_{gd} . 这样的策略增加了算法在迭代末期的局部查找能力.

跟踪实验结果发现 ECF 的使用, 增加了算法查找最优解的能力. 但是 ECF 也限制了 BSS 中部分 SO 发挥作用. 因此降低了粒子的多样性. 从而算法更容易早熟. 为了避免早熟, ECFG-PSO 算法加入了二次初始化策略. 如果算法经过多次迭代, 其解不更新, 则认为算法早熟 (实验中 7 次迭代解不更新, 则认为算法早熟), 此时将部分粒子二次初始化, 从而避免了算法过早的陷入早熟.

3.4 算法描述

在算法迭代初期, 程序还没有收敛. 所以这个时候生成 ECF 是不理想的, 下面以 Q 次迭代中在 $Q/2$ 的时候生成 ECF 例描述算法, T 记录当前的迭代次数, γ 为染色体符合度.

Step1:

粒子群规模为 M, TSP 中城市个数为 n. 算法随

机生成规模为 M 的粒子、 p_{id} 、 p_{gd} 、优秀染色体长度 $\lambda \in (1, n)$ 、符合度 $\gamma \in (0, 1)$;

Step2:
IF ($T \geq Q/2$) 转 Step4;
Step3:
对于每个粒子计算 v_i ,更新 x_i
3.1 $P_1 = p_{id} - x_{id}$;
3.2 $P_2 = p_{gd} - x_{id}$;
3.3 $v_i = v_i \cdot \text{rand}() \cdot P_1 \oplus \text{rand}() \cdot P_2$;
3.4 $x_i = x_i + v_i$;转 Step5;
Step4:
4.1 生成 ECF;
IF (ECF 不存在) 转 Step3;
对于每个粒子计算 v_i ,更新 x_i
4.2 $P_1 = p_{id} - x_{id}$;
4.3 $P_2 = p_{gd} - x_{id}$;
4.4 $v_i = v_i \cdot \text{rand}() \cdot P_1 \oplus \text{rand}() \cdot P_2$;
4.5 使 ECF 作用于 $x_i = x_i + v_i$; 转 Step5;
Step5:

判断是否满足结束条件(找到最优解或者迭带次数到达 Q), 如果不满足,转到 Step2.

4 实验结果

文中采用了 TSP 的经典模型 Oliver30 验证算法的有效性. 实验环境为: Pentium4 2. 93GHz CPU, 256MB RAM, WinXP OS, VC++ 6. 0. 算法中, $M = 200, Q = 500$, 当迭代次数时 $T = 10$, ECF 开始发挥作用. $\lambda = 6, \gamma = 0.8$. 限迭代 60 次. 文[4]中 ACO 算法得出的 Oliver30 的最优解为 423. 74, 这是 423. 74 是目前各种算法应用于 Oliver30 问题所能得出的最优解. ECFG-PSO 得出的最优解同样为 423. 74.

表 1 记录了在相同的实验环境下对 DPSO 和 ECFG-PSO 算法连续运行 10 次所得得最优解、平均解和错误率. 错误率的计算方法:

$$\text{Err} = \frac{\text{Ave} - 423.74}{423.74} \times 100\%$$

(5)

实验对 DPSO 和 ECFG-PSO 算法连续运行 10 次实验结果表 1 所示.

表 1 DPSO 与 ECFG-PSO 算法比较

算法 (Algorithm)	群规模 (M)	连续运行 10 次所得解										平均解 (Ave)	错误率 (Err)
		Run1	Run2	Run3	Run4	Run5	Run6	Run7	Run8	Run9	Run10		
ECFG-PSO	200	423.74	423.74	424.74	423.74	423.74	423.74	424.90	423.74	423.74	423.74	423.96	0.05%
DPSO	500	432.66	423.74	425.27	425.48	423.95	423.74	426.31	423.74	423.74	423.74	424.34	0.14%

图 1 中两条曲线分别记录了 DPSO 和 ECFG-PSO 算法的十次实验中同一次叠代后的平均解.

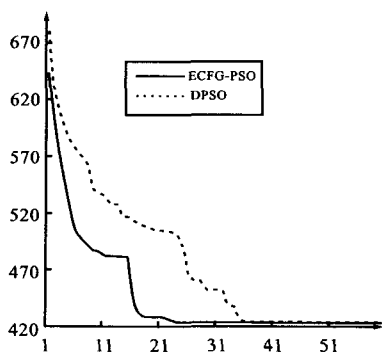


图 1 DPSO 和 ECFG-PSO 收敛曲线

从表 1 和图 1 可以看出, 在种群规模较小的情况下, 对同一 TSP 问题求解其收敛速度快于 DPSO 算法. 并且 ECFG-PSO 算法稳定性较好, 求解精度较高.

5 结束语

文中将 DPSO 算法与 GA 算法有机结合, 提出 ECFG-PSO 算法. 从实验结果看算法在性能方面

相对于 DPSO 算法具有一定的改进. 但是从实验结果分析来看, 算法在收敛速度上仍然有待提升. 目前算法合适的参数配置, 以及 ECF 发挥作用的时机等问题仍有待进一步研究.

参考文献:

[1] 李朝锋, 居红云, 王琪. 基于 QPSO 的模糊 C 均值聚类算法[J]. 微电子学与计算机, 2008, 25(7): 195-197.
[2] 徐文龙, 孙俊, 须文波. 基于干扰因子的 QPSO 算法改进[J]. 微电子学与计算机, 2008, 25(4): 129-130.
[3] Kennedy J, Eberhart R. Particle swarm optimization[C]// Proceedings of the 1995 IEEE International Conference on Neural networks. Perth. AUST: IEEE, 1995(44): 1942-1948.
[4] Goldberg DE. Genetic Algorithms in search, Optimization, and machine learning[M]. Boston: Addison - Wesley, 1989.
[5] 郝晋. 求解复杂 TSP 问题的随机扰动蚁群算法[J]. 系统工程理论与实践, 2002(9): 88-92.

(下转第 209 页)

结构加固的锁存器电路所能达到的抗单粒子效应的阈值 LET 值为 $3.5\text{Mev}\cdot\text{cm}^2/\text{mg}$. 而在 $0.18\mu\text{m}$ 工艺下,未加固的锁存器单元电路的阈值 LET 值在 $0.3\sim 0.7\text{Mev}\cdot\text{cm}^2/\text{mg}$ 之间^[6]. 显然,采用 DICE 结构的锁存器的抗单粒子效应的能力得到了提高.

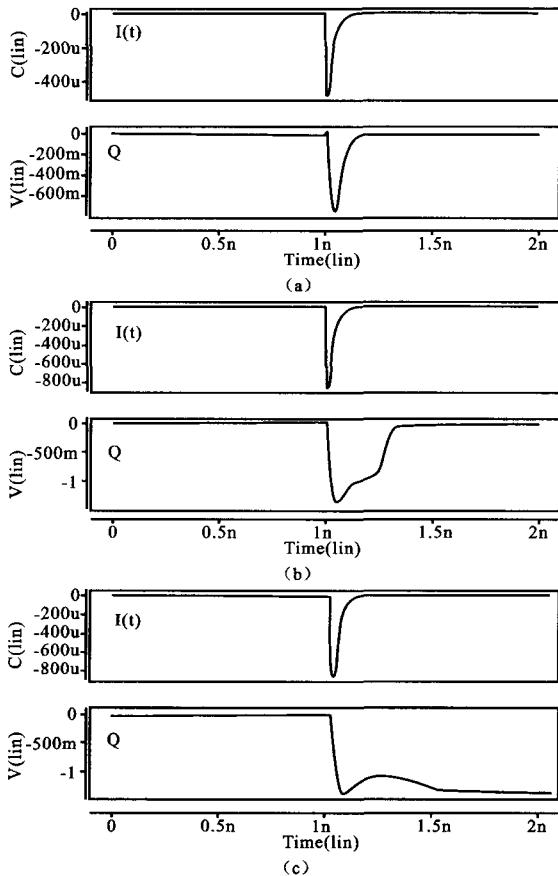


图5 D-Latch 抗单粒子效应模拟

5 结束语

文中研究了基于 RHBD 技术的 DICE 结构电路,并将 DICE 结构电路用于 D-Latch 的设计中,绘制版图并提取了版图 RC 寄生参数,做了带寄生参数的功能仿真;对其抗单粒子效应进行了模拟仿真. 仿真结果表明加固后的锁存器电路抗单粒子效

应能力得到了提高. 文中并没有考虑版图设计对抗辐射能力的影响,实际上,电路设计结合版图上的优化可以达到更优越的抗辐射能力. 同时,对于功耗、面积、速度的综合考虑在抗辐射加固电路的设计中也是非常重要的.

参考文献:

- [1] Balasubramanian A, Bhuva B L, Black J D, et al. RHBD techniques for mitigating effects of single-event hits using guard-gates[J]. IEEE Transactions on Nuclear Science, 2005, 52(6): 2531-2535.
- [2] Micheal Nicolaidis. Design for soft error mitigation[J]. IEEE Transactions on Device and Materials Reliability, 2005, 5(3): 405-417.
- [3] Jonathan E Knudsen, Lawrence T Clark. An area and power efficient radiation hardened by design flip-flop[J]. IEEE Transactions on Nuclear Science, 2006, 53(6): 3392-3399.
- [4] Rockett L. An SEU hardened CMOS data latch design[J]. IEEE Trans. Nucl. Sci., 1988, 35(6): 1682-1687.
- [5] Calin T, Vargas F, Nicolaidis M, et al. A low-cost, highly reliable SEU-tolerant SRAM: Prototype and test results[J]. IEEE Trans. Nucl. Sci., 1995, 42(6): 1592-1598.
- [6] Nadim Haddad, Leonard Rockett, Scott Doyle. Design considerations for next generation radiation hardened SRAMs for space applications [C]// Aerospace, 2005 IEEE Conference. Big Sky, 2005.
- [7] 李玉虹,赵元富,岳素格. $0.18\mu\text{m}$ 工艺下单粒子加固锁存器的设计与仿真[J]. 微电子学与计算机, 2007, 24(12): 66-69.

作者简介:

胡明浩 男, (1984-), 硕士研究生. 研究方向为抗辐射加固技术、数字 IC 设计.

李 磊 男, (1982-), 博士. 研究方向为数字 IC 设计、NoC 设计.

饶全林 男, (1986-), 硕士研究生. 研究方向为抗辐射加固技术、数字 IC 设计.

(上接第 205 页)

- [6] Kennedy J, Eberhart R C. A discrete binary version of the particle swarm algorithm[C]// Proceedings of IEEE Conference on Systems, Man, and Cybernetics. Orlando, 1997 (5): 4104-4108.
- [7] 黄岚,王康平. 粒子群优化算法求解旅行商问题[J]. 吉林大学学报, 2004(4): 477-480.

- [8] 梁艳春,冯大鹏. 遗传算法求解旅行商问题时的基因片段保序[J]. 系统工程理论与实践, 2004(4): 7-13.

作者简介:

程 乐 男, (1979-), 讲师. 研究方向为人工智能.

张洪斌 男, (1961-), 研究员. 研究方向为系统仿真.