



(12) 发明专利

(10) 授权公告号 CN 1533552 B

(45) 授权公告日 2011.07.13

(21) 申请号 02811225.3

(72) 发明人 A·L·布查克 H·王

(22) 申请日 2002.04.04

(74) 专利代理机构 中国专利代理(香港)有限公司 72001

(30) 优先权数据

代理人 陈景峻 王勇

60/282,366 2001.04.06 US

09/893,108 2001.06.27 US

(51) Int. Cl.

G06N 3/12(2006.01)

(85) PCT申请进入国家阶段日

2003.12.03

(56) 对比文件

(86) PCT申请的申请数据

PCT/US2002/010477 2002.04.04

BUCZAK A L ET AL. BUCZAK A L ET AL: "Genetic algorithm convergence study for sensor network optimization". INFORMATION SCIENCES. 2001, 268.

(87) PCT申请的公布数据

WO2002/082371 EN 2002.10.17

审查员 伊健

(73) 专利权人 霍尼韦尔国际公司

地址 美国新泽西州

权利要求书 3 页 说明书 15 页 附图 9 页

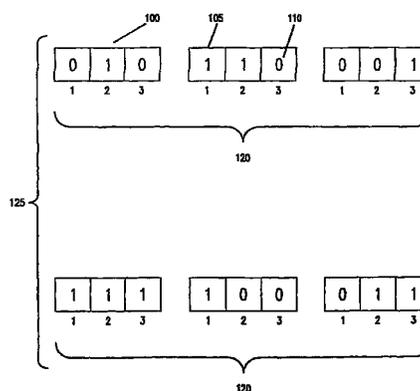
(54) 发明名称

基因算法优化方法

(57) 摘要

本发明包含一种用于从一个用于对至少一个目标进行跟踪的传感器网络中选择传感器的方法,该方法具有以下步骤:定义具有 n 个染色体的基因算法结构的个体,其中每个染色体代表一个传感器,基于想要的跟踪的特性定义适合性函数,选择个体中的一个或多个以便将其包含进一个初始总体,对该初始总体执行一种基因算法直到所定义的趋同标准被满足,其中对基因算法的执行具有以下步骤:从该总体中选择最适合的个体,从该总体中选择随机个体以及通过该最适合的个体和随机选择的个体创造后代。本发明的另一种实施方案包含另一种用于从一个用于对至少一个目标进行跟踪的传感器网络中选择传感器的方法,该方法具有以下步骤:定义具有 n 个染色体的基因算法结构的个体,其中每个染色体代表一个传感器,基于想要的跟踪的特性定义适合性函数,选择个体中的一个或多个以便将其包含进一个初始总体,对该初始总体执行一种基因算法直到所规定的趋同标准被满足,其中对基因算法的执行具有以下步骤:从该总体中选择最适合的个体,以及通过该最适合的个体创造后代,其中仅通过突变来实现对后代的创造,其中在任何一次突

变期间仅有 i 个染色体发生突变,并且其中 i 的值为从 2 到 n-1。本发明还包含一种用于跟踪目标的包含 N 个传感器的传感器网络,一种用于 N 个传感器与控制器进行通信的装置和一种通过利用本发明的其中一种方法而具备控制和管理该 N 个传感器的能力的控制器。



1. 一种用于从用于跟踪至少一个目标的传感器网络中选择传感器并用计算机实现的方法,所述用计算机实现的方法包含以下步骤:

(a) 定义具有 n 个染色体的基因算法结构的个体,其中每个染色体代表一个传感器;

(b) 根据想要的跟踪的特性,定义适合性函数;

(c) 选择所述个体中的一个或多个,以便将其包含在初始总体中;

(d) 对所述总体执行基因算法,直到所定义的收敛标准被满足,其中对所述基因算法的执行包含以下步骤:

(i) 基于所述适合性函数从所述总体中选择最适合的个体;

(ii) 从所述总体中选择随机个体;以及

(iii) 从所述最适合的个体和随机选择的个体创造后代;

(e) 选择传感器,其中包含有所定义的收敛标准被满足时而存在的总数量的所述个体是所选择的传感器。

2. 如权利要求 1 所述的用计算机实现的方法,其中代表所述传感器的所述染色体包含所述传感器的二进制或实数标识。

3. 如权利要求 1 所述的用计算机实现的方法,进一步包含将个体定义为包含 n 个染色体,其中 n 是通过将跟踪所述目标所必需的传感器的数量与要被跟踪的所述目标的数量相乘得到的。

4. 如权利要求 1 所述的用计算机实现的方法,其中步骤 (b) 的所述想要的特性包含最小功率消耗。

5. 如权利要求 1 所述的用计算机实现的方法,其中步骤 (b) 的所述想要的特性包含最小跟踪误差。

6. 如权利要求 1 所述的用计算机实现的方法,其中步骤 (b) 的所述想要的特性包含最小功率消耗和最小跟踪误差。

7. 如权利要求 6 所述的用计算机实现的方法,其中步骤 (b) 的所述适合性函数包含公式:

$$F = -\left(w_1 \sum_{i=1}^n E_i + w_2 \sum_{j=1}^m P_j \right)$$

其中 E_i ($i = 1, 2, \dots, n$) 是为跟踪第 i 个目标而估计的位置误差,其中 P_j ($j = 1, 2, \dots, m$) 是第 j 个传感器的功率消耗值; n 是目标的数量; m 是所选择的传感器的总数量,并且 w_1 和 w_2 是两个加权常数。

8. 如权利要求 1 所述的用计算机实现的方法,其中步骤 (c) 中的对所述个体的所述选择是通过随机方法来实现的。

9. 如权利要求 1 所述的用计算机实现的方法,其中步骤 (d) 的所述收敛标准包含特定数量的代。

10. 如权利要求 1 所述的用计算机实现的方法,其中步骤 (d) 的所述收敛标准包含特定数量的代,经过这些特定数量的代之后,所述总体中的最适合的个体不会再有改进。

11. 如权利要求 1 所述的用计算机实现的方法,其中通过轮盘选择、比赛选择、随机数生成,或者它们的组合来实现在步骤 (d) 中从所述总体中选择所述随机个体。

12. 如权利要求 1 所述的用计算机实现的方法,其中通过突变、交叉,或者它们的组合来实现步骤 (d) 中的所述后代的所述创造。

13. 如权利要求 12 所述的用计算机实现的方法,其中通过突变、交叉,或者它们的组合来实现步骤 (d) 中的所述后代的所述创造,并且在任意一次突变或交叉中仅有 i 个染色体受到影响,其中 i 的值从 2 到 $n-1$ 。

14. 如权利要求 13 所述的用计算机实现的方法,其中 i 的值为 2。

15. 一种用于从用于跟踪至少一个目标的传感器网络中选择传感器的设备,包含:

(i) 用于定义具有 n 个染色体的基因算法结构的个体的装置,其中每个染色体代表一个传感器;

(ii) 用于基于想要的跟踪的特性定义适合性函数的装置;

(iii) 用于选择所述个体中的一个或多个以便将其包含在初始总体中的装置;

(iv) 用于对所述总体执行基因算法直到所定义的收敛标准被满足的装置,其中所述用于对所述总体执行基因算法直到所定义的收敛标准被满足的装置包含:

(a) 基于所述适合性函数从所述总体中选择最适合的个体的部件;

(b) 从所述总体中选择随机个体的部件;以及

(c) 通过所述最适合的个体和被随机选择的个体创造后代的部件;

(iv) 用于选择传感器的装置,其中包含有所定义的收敛标准被满足时而存在的总数量的所述个体是所选择的传感器。

16. 如权利要求 15 所述的设备,其中代表所述传感器的所述染色体包含所述传感器的二进制或实数标识。

17. 如权利要求 15 所述的设备,进一步包含用于将个体定义为包含 n 个染色体的装置,其中 n 是通过将跟踪所述目标所必需的传感器的数量与要被跟踪的所述目标的数量相乘得到的。

18. 如权利要求 15 所述的设备,其中所述想要的特性包含最小功率消耗。

19. 如权利要求 15 所述的设备,其中所述想要的特性包含最小跟踪误差。

20. 如权利要求 15 所述的设备,其中所述想要的特性包含最小功率消耗和最小跟踪误差。

21. 如权利要求 20 所述的设备,其中所述适合性函数包含公式:

$$F = - \left(w_1 \sum_{i=1}^n E_i + w_2 \sum_{j=1}^m P_j \right)$$

其中 E_i ($i = 1, 2, \dots, n$) 是为跟踪第 i 个目标而估计的位置误差,其中 P_j ($j = 1, 2, \dots, m$) 是第 j 个传感器的功率消耗值; n 是目标的数量; m 是所选择的传感器的总数量,并且 w_1 和 w_2 是两个加权常数。

22. 如权利要求 15 所述的设备,其中对所述个体的所述选择是通过随机方法来实现的。

23. 如权利要求 15 所述的设备,其中所述收敛标准包含特定数量的代。

24. 如权利要求 15 所述的设备,其中所述收敛标准包含特定数量的代,经过这些特定数量的代之后,所述总体中的最适合的个体不会再有改进。

25. 如权利要求 15 所述的设备,其中通过轮盘选择、比赛选择、随机数生成,或者它们的组合从而实现从所述总体中选择所述随机个体。

26. 如权利要求 15 所述的设备,其中通过突变、交叉,或者它们的组合来实现所述后代的所述创造。

27. 如权利要求 15 所述的设备,其中通过突变、交叉,或者它们的组合来实现所述后代的所述创造,并且在任意一次突变或交叉中仅有 i 个染色体受到影响,其中 i 的值从 2 到 $n-1$ 。

28. 如权利要求 15 所述的设备,其中 i 的值为 2。

基因算法优化方法

[0001] 本发明请求于 2001 年 4 月 6 日提交的题为“基因算法优化方法”的美国临时申请 No. 60/282366 的优先权,该申请所公开的全部内容被并入本申请中。

发明领域

[0002] 本发明总体上涉及改进的优化方法。特别地,本发明涉及基因算法并且可以用于对高度多模式和欺骗函数进行优化,其一个例子为选择一个传感器网络的传感器个体用于跟踪一个特定目标。

[0003] 发明背景

[0004] 对具有多个独立变量的高度多模式和欺骗函数进行优化是非常耗费时间的,这是因为这些函数所展现出的大的搜索空间和多个最适合条件。通常,函数具有越多的独立变量,优化过程就越难实现。

[0005] 特别难以优化的函数通常共享某种特性,这些特性包含:多模式、不可微分性、不连续性、特征类型(不规则)变量、以及大量的独立变量。这种函数的经典的数学例子包含例如 Rastringin 的函数、欺骗函数、Holland 的 Royal Road 函数。

[0006] 还存在许多实际的情况,在这些情况中,由高度多模式和 / 或欺骗函数来表示问题。这种实际情况的例子包含在计算机 / 无线网络中对路由器的选择、对芯片上晶体管的组织、例如蛋白质折叠和 RNA 折叠的生物计算应用、可展开的硬件、工作车间调度和维护调度问题、制定时间表、通过传感器网络跟踪目标、传感器部署计划工具以及对传感器网络的控制和管理。对传感器网络的控制和管理将作为一个示例性的大型多模式的实际问题被进一步考虑。

[0007] 无人看管的地面传感器 (“UGS”) 可以极大地增加军事行动的有效性和能力。在市场上最容易买到的 UGS 是独立工作的多功能、集成传感器平台。UGS 的一个例子是一种声学 UGS,该声学 UGS 包含三个声学麦克风(用于实现准确的方位角测量)、一个地震传感器、一个磁传感器、一个全球定位传感器、一个方位传感器、集成通信和信号处理电子器件,以及一个电池。这样一个平台的体积通常约为 1ft^3 (28320cm^3),并且价格非常昂贵。因为存在这些缺点,这种 UGS 通常不被用来支持用于小的、快速可部署的军事行动的远程监视应用。

[0008] 对这种较大的、昂贵的传感器平台的一种替换的方案是使用大约为 2in^3 (大约 33cm^3) 的小型 UGS,这种小型 UGS 比较便宜并且可方便地由单个战斗机对其进行部署。较小的传感器,例如那些在这种小型 UGS 中使用的传感器,通常只能在较短范围内进行通信和目标感应,并且可能仅可以感应单个目标特性(如地震振动或化学探测)。另外,较小的传感器通常具有较短的工作寿命,这是由于电池较小的缘故。由于这些特性,为了实现与较大的 UGS 所能实现的目标相同的目标,不得不部署更多的这种小 UGS。但是,单独进行工作的单个小型 UGS 将不能执行监视任务。

[0009] 对这种问题的一种替换的方案是将这种小的、低造价的 UGS “散播”在监视区域,并且使这些传感器可以对它们自己进行组织并协同工作。如这样的一种 UGS 网络将会具有体积较大的单个进行工作的传感器所不具有的许多优点。例如,位于中间的 UGS 可以为位

于较远位置的传感器充当“短距离”通信中继站。一个网络中具有更多传感器可以允许具有不同类型的传感器,这将使得网络的联合操作具备更广泛的功能。另外,网络中内置的冗余将会使得该网络受单点故障和 / 或传感器信号丢失的影响较小。

[0010] 为了使具有多个小的、便宜的 UGS 的网络能够以可以接受的方式进行工作,必须开发出一种用来组织和控制这样一种网络的算法和方法。以下的问题被认为是一个没有唯一的解决方案的多目标优化问题:选择一个传感器的最佳集合用来对进入监视区域的目标进行探测、跟踪以及分类,而与此同时使该传感器网络的功率消耗最小化。另外,如果目标或传感器数目是按线性规律增加的,则优化会导致组合的搜索空间按照指数规律增加。

[0011] 美国专利 No. 6055523 (Hillis) 公开了一种用于在使用一个或多个传感器进行的多目标跟踪中分配传感器报告的方法。该方法通过多次时间扫描从至少一个传感器接收传感器报告,使用公式将基因算法总体中的个体表示为这些传感器报告的排列,并且然后使用标准基因算法技术来发现被跟踪的目标的路径。该方法使用基因算法来确定被跟踪的目标的路径,而不是选择传感器或传感器报告以进行使用。

[0012] 因此,需要一种改进的算法,该算法可以为了完成对性能的多个不同变量同时进行优化的目的而从网络中选择单个的传感器。

[0013] 发明概述

[0014] 根据本发明,提供了一种用于从一个传感器网络中选择传感器以用于对至少一个目标进行跟踪的方法,该方法具有以下步骤:定义具有 n 个染色体的基因算法结构的个体,其中每个染色体代表一个传感器,根据想要的跟踪的特性来定义适合性函数,选择一个或多个个体并将其包含进一个初始总体,对该初始总体执行一种基因算法直到所规定的趋同 (convergence) 标准被满足,其中基因算法的执行具有以下步骤:从该总体中选择最适合的个体,从该总体中选择随机个体以及通过该最适合的和随机选择的个体来创造后代。

[0015] 根据本发明的另一种实施方案,提供了一种用于从一个传感器网络中选择传感器以用于对至少一个目标进行跟踪的方法,该方法具有以下步骤:定义具有 n 个染色体的基因算法结构的个体,其中每个染色体代表一个传感器,根据想要的跟踪的特性来定义适合性函数,选择一个或多个个体并将其包含进一个初始总体,对该初始总体执行一种基因算法直到所规定的趋同标准被满足,其中对基因算法的执行具有以下步骤:从该总体中选择最适合的个体,以及通过该最适合的个体创造后代,其中仅通过突变来实现对后代的创造,其中在任何一次突变期间仅有 i 个染色体发生突变,并且其中 i 的值为从 2 到 $n-1$ 。

[0016] 根据本发明的另一种实施方案,提供了一种用于跟踪目标的传感器网络,包含 N 个传感器、一种用于 N 个传感器与控制器进行通信的装置和一个可以通过利用根据本发明的方法从而控制和管理该 N 个传感器的控制器。

[0017] 优选地,通过突变、交叉 (crossover) 或者它们的组合来创造后代。更加优选地,仅通过突变来实现对后代的改变。

[0018] 优选地,后代的改变发生在 i 个染色体,其中 i 的值从 2 到 $n-1$,其中 n 是组成一个染色体的基因的数目。更加优选地, i 的值为 2。

[0019] 附图简述

[0020] 图 1 描述了一个基因算法的总体的一般结构。

[0021] 图 2 描述了代表一个基因算法中的步骤的普遍化的流程图。

- [0022] 图 3a 描述了一种单点、单染色体交叉。
- [0023] 图 3b 描述了一种双点、单染色体交叉。
- [0024] 图 4a 描述了一种突变,其中因为突变的概率,仅有一个基因发生了突变。图 4b 描述了一种突变,其中因为突变的概率,两个基因发生了突变。
- [0025] 图 5 描述了一种根据本发明的单点、 C_2 交叉。
- [0026] 图 6 描述了一种根据本发明的 C_2 突变。
- [0027] 图 7 描述了一种基因算法的结构,该基因算法用于为目标跟踪 / 标识选择最佳传感器的过程。
- [0028] 图 8 描述了一种代表根据本发明的一个方面的方法的普遍化的流程图,该方法用于控制和管理一个传感器网络。
- [0029] 图 9 描述了八种算法在使传感器控制优化方面的性能的平均最佳适合性。
- [0030] 图 10 描述了图 9 中所表示的五种算法所进行的优化所必需的时间和有效性。
- [0031] 图 11 描述了图 10 中所描述的五种算法改进百分比随时间发生变化的情况。
- [0032] 优选实施方案详述
- [0033] 发明装置
- [0034] 根据本发明的一种装置包含至少一个传感器、一个处理器,以及一种基因算法。
- [0035] 术语“实体”将被用于对本发明的通篇描述。术语实体应该被理解为广泛地包含多种不同的电子术语,例如任何被用于感应目标或者可以被用于感应目标的传感器,或者在计算机或无线网络中的路由器。实体一般是指,例如可以被用来探测目标的一种特性的传感器。这种特性的例子包含速度、位置、方位、类型(或者标识)、大小。本发明并不限于任意特定类型或数量的传感器。尽管一种优选的实施方案包含小的、便宜的传感器,通篇使用的实体这一术语并不受此限制。可替换地,术语实体还可以指从任意类型的实体(例如一个传感器)中接收到的数据。
- [0036] 优选地,用于本发明的一种实施方案的传感器是这样一种传感器:它小于大约 2in^3 (大约 33cm^3),它的生产和运行成本都较低,并且可以被容易地部署。这样一种传感器几乎可以是任意类型的传感器,包含但并不限于声学传感器、地震传感器、机械传感器,或者半导体激光传感器。许多公司生产的传感器都可以被用于本发明的一种实施方案,这样的公司的例子包含但不限于 Northrop-Grumman、SenTech、Raytheon、BAE、Aliant and Rockwell Sciences Center。
- [0037] 术语“网络”是指多于一个的传感器,这些传感器可以与其它传感器进行通信并且受一个或多个系统或处理器的控制。网络中的一些传感器可能并不能被使用,例如因为它们可能在范围之外,或者它们的电池没电了,或者可能仅仅是它们未被使用但仍然被认为是网络的一部分。网络中的传感器之间的通信可以通过有线或无线方式来实现。只要存在用于控制传感器的一个单独的计划或方法,一个单个的处理器或者多个不同的处理器就可以控制该网络。
- [0038] 术语“处理器”是指一个或多个设备,这一个或多个设备可以确定如何控制和管理传感器以及实际地对传感器进行控制和管理。通常,这种处理器包含可以执行本方法的必要的步骤并可以控制网络的单个传感器的任意可以被使用的处理系统。可以执行处理器功能的处理系统的例子包含但不限于一种 500MHz Compaq 膝上型计算机。可以理解,控制可

编程计算机的软件程序、基于硬件的装置都可以替换地实施本方法并且是本发明的设备的一部分,其中基于硬件的装置包含通用的或由客户设计的集成电路设备,这些通用的或由客户设计的集成电路设备包含集成电路微处理器和永久指令保持存储器。

[0039] 术语“目标”是指正在被跟踪的物体、动物,或者人类。优选地,正在被跟踪的目标为一个物体,例如一个陆地或空中交通工具。通常,对传感器进行配置从而获得关于该目标的一些类型的信息。这些信息可以包含但不限于该目标的大小、标识、速度,以及方位。

[0040] 术语“感应”或者“被感应”是指获得关于目标的一些信息随时间而发生变化的情况的过程。通过感应而获得的这些信息包含但不限于经典的通过跟踪、取平均值从而获得的目标位置随时间发生变化的信息。这种位置通常为 2 维 x 、 y 坐标,或者 3 维 x 、 y 、 z 坐标。感应还包含获得其它关于标识的信息,例如目标的一些物理特性。

[0041] 基本基因算法

[0042] 本发明的方法和设备使用改进的基因算法。为了理解该改进的基因算法,首先将对基本基因算法和它们的术语进行讨论。

[0043] 基因算法是基于自然选择和遗传学的搜索算法。通常来讲,它们将适者生存的概念和信息的随机化交换结合在一起。每个基因算法代中都有一个包含个体的总体。这些个体可以被看作正被解决的问题的候选解决方案。在每个连续的代中,使用上一代的最适合的部分从而创造一个新的个体集合。但是,还要非经常性地将随机化新信息包含进来,以便使重要数据不会丢失和被忽视掉。

[0044] 图 1 举例说明了基因算法所基于的结构。基因算法的一个基本概念是:它就一个总体中的个体而言对一个问题可能的解决方案进行了定义。染色体 100,也被称作位字符串,包含多个基因 105,其中该基因 105 也被称作特征、特性、或者位。每个基因 105 具有一个等位基因,或者可能值 110。一个特定的基因 105 还具有一个位置或者字符串位置 115,该位置或者字符串位置 115 表示基因在染色体 100 中的位置。

[0045] 在一个正在运行的基因算法中,通过对问题的可能的解决方案进行编码从而确定染色体 100。例如,考虑到达一个特定目的地的可能的路线以及完成每种路线所需的时间。多种因素将会决定任意一种特定的路线将会花费多长的时间,这些因素中的一些因素包含,例如:路线的长度、该路线上的交通流量状况、该路线上的路面状况,以及该路线上的天气情况。可以通过为这些因素(或者基因 105)的每个因素赋值(或者同位基因 110)从而为每个路线构建染色体 100。

[0046] 一个基因型,也被称作结构或者个体 120,可以包含一个或多个染色体 100。在图 1 中,一个基因型 120 包含 3 个单独的染色体 100。依此类推,如果问题包含用于包含多个路段的总路程的多种可能的路线,那么就存在具有多于一个染色体 100 的基因型或者个体 120。该总路程的每个路段将会有一座城市(或者染色体 100)。一组个体 120 组成总体 125。总体 125 中的个体 120 的数量(所谓的总体大小)取决于正在被解决的特定的问题。

[0047] 在前面已经解释了基因算法运行所基于的结构,接下来将讨论这些基因算法运行的方式。图 2 描述了一种基因算法的运行情况。

[0048] 第一步是初始化步骤 150。初始化是通过由操作基因(operator)指定涉及基因算法运行的方式的一些细节来实现的。需要在初始化步骤 150 被指定或者被选择的细节包含,例如总体大小、特定的操作基因发生的概率,以及对最终解决方案的预期。初始化所必

需的细节部分地取决于基因算法的准确的运行。在初始化时被选择的参数可以使用基因算法支配为了确定想要的解决方案所必需的时间和资源。还应该理解,该初始化步骤 150 是可选的,这是因为通过初始化步骤 150 获得的所有信息可以被包含在算法自身当中,并且在初始化步骤期间可以不用要求用户输入。

[0049] 基因算法中的下一步是对初始总体的选择步骤 155。对初始总体的选择通常是通过个体 120 的随机选择来实现的,但是也可以通过其它方法来实现。组成初始总体的个体 120 的数量部分地是由在初始化步骤 150 所选择的参数来确定的。通常,使用一个随机数发生器通过为每个染色体 100 中的每个基因 105 确定值 110 从而创建初始总体。

[0050] 接下来,被随机选择的总体的个体 120 的适合性在适合性的确定步骤 160 中被确定。个体 120 的适合性取决于指派基因算法对其进行优化的特定的问题。例如,适合性可能取决于个体 120 的成本、个体 120 对于特定任务的有效性,或者它们的组合。个体 120 的适合性必须可以定量地被测量和确定,例如使用一个公式。总体中的每个个体 120 都具有一个特定的适合性值。

[0051] 下一步是检查趋同标准是否已经被实现的步骤 165。在经典的基因算法中,这一步骤通常指在进行检查的时候查看个体的适合性是否满足某些定义的适合性标准。通常,在实际的应用中,适合性的可能的或者可接收的水平可能是未知的,所以基因算法例如在一些代后被停止,或者在其中最适合的个体不发生变化的一些代后被停止。在以上两种情况的任意一种情况中,这一步骤检查从而确定要求,即代的数量或是总体的适合值,是否已经得到了满足。任意给定的总体要么会满足该标准,要么不会满足该标准。如果总体满足了趋同标准,该总体将会被认为是用来跟踪目标的传感器的最佳总体,也就是最终总体。在这种情况下,接下来的步骤是最终总体的输出步骤 185。最终总体的输出可以通过多种不同的方式来实现,包括但不限于将最终总体的属性打印为一种硬复制版本、将最终总体的属性保存为电子格式,或者使用最终总体来控制或管理一些过程。

[0052] 如果检查趋同标准是否已经被实现的步骤 165 显示该总体并没有满足所要求的标准,那么接下来的步骤是交配集合选择步骤 170。基因算法中的交配集合选择步骤 170 可以通过多种方式来实现,但是通常部分地基于所涉及的个体的适合性。例如,可以使用偏置的轮盘来选择个体,其中该偏置基于个体的适合性。用来选择交配集合的另一种方法严格地基于适合性值;总体中的一定比例的最适合个体被选择从而进行交配。还有另一种方法使用比赛选择,首先,随机选择 k 个个体 120。然后,确定每 k 个元组中最适合的个体 120,并且这些个体被复制进交配集合。

[0053] 接下来的步骤是后代的创造步骤 180。在该步骤中,对在交配集合选择步骤 170 中选择的双亲进行有修改或无修改地组合以创造后代。并不是交配集合所创造的每个成员都要在后代的创造步骤 180 中被修改。通常,交配集合的一个特定的成员是否要被修改是由概率确定的。例如,这些概率既可以在初始时被指定,也可以通过从交配总体或交配对中获得的信息来确定。可以通过多种方式来实现对后代的修改,所述方式被称作操作基因。通常,按照给定的概率对交配集合的成员施加操作基因。通常所使用的操作基因包括但不限于交叉、突变、倒位、优势改变、分离和易位,以及染色体复制。这里仅解释交叉和突变。

[0054] 交叉是这样一种方法,通过该方法,两个不同的染色体 100 上的基因 105 被分散到这两个染色体 100 之间。单点交叉是通过沿染色体 100 随机选择一个位置 k 来实现的,该

位置位于 1 和染色体长度减 1 之间。两个后代是通过交换位于位置 $k+1$ 和染色体 100 的全长之间的所有基因 105 被创造出来的。有多种不同类型的交叉, 包含但不限于单点、双点、一致 (uniform)。还可以在一个个体 120 的一个或多个染色体 100 上实现交叉。通常, 仅在一个染色体上或在每个染色体上实现交叉。

[0055] 图 3a 举例说明了一种单点、单染色体交叉。在两个未经修改的后代个体 120 上选择一个交叉点 130。位于包含交叉点 130 的基因 105 之内的等位基因 110 被交换到交叉点 130 之后。基因 105 仅在该染色体 100 上被交换。在进行完交叉之后, 创造出了被修改的后代个体 120'。图 3b 举例说明了一种双点、单染色体交叉。在双点、单染色体交叉中, 交叉点 130 和第二交叉点 132 是在同一个染色体 100 内随机选择的。在这种交叉中, 在到达第二交叉点 132 之前, 在一个染色体 100 之内位于交叉点 130 之后的等位基因 110 被交换, 在第二交叉点 132 处, 等位基因 110 保持它们和在最初的染色体 100 中相同。理论上, 在任意一个染色体中, 有多少个基因 105 就可以选择多少个交叉点。

[0056] 突变是这样一种方法, 通过这种方法, 染色体 100 上的一个或多个基因 105 被修改。按照通常在基因算法的初始化步骤中被确定的突变的概率来选择用于突变的基因 105。在一次事件中可以对一个染色体 100 上的多于一个基因 105 进行突变。突变的概率通常比交叉的概率要低得多。突变通常被认为是用来确保有用的基因不会丢失的一种方法。在一个或多于一个染色体 100 上可以发生多个突变。可以发生突变的染色体 100 的数量的范围是从 1 到 n , 其中 n 是在一个个体 120 中的染色体 100 的数量。

[0057] 图 4a 表示了一种单染色体突变。基因 105 处的占据突变点 140 的等位基因 110 被改变为某种其它的等位基因 110。在一种二进制编码中, 突变是将 0 变为 1, 或者将 1 变为 0。因为通常发生突变的概率比较低, 所以某些基因发生突变, 而某些基因不发生突变。在后代的创造步骤 180 之后, 适合性的确定步骤 160 被重复, 在适合性的确定步骤 160 之后跟有检查是否已经实现趋同标准的步骤 165。如果总体没有满足标准的话, 该循环就一直进行。如上述, 如果总体满足了趋同标准, 就执行输出步骤 185 并且算法被完成。

[0058] 改进的基因算法

[0059] 为了解决例如传感器网络的控制和管理的多模式问题, 本发明包含改进的基因算法。在前面进行的对基本的基因算法的讨论形成了在这里所提供的改进的算法的基础。本发明使用了三种单独的改进。这些改进可以被分别用于基本的基因算法, 可以被共同用于基本的基因算法, 可以被用于非基本的基因算法, 或者以上这几种情况的组合。

[0060] 在本发明中使用的第一种改进被称作 C_i 交叉。 C_i 交叉描述了这样一种交叉的发生, 该交叉恰好影响一个个体 120 中的 i 个染色体 100。每个交叉可以为任意类型的交叉, 包含但不限于单点、多点, 或者一致。单点交叉是当基因材料, 也就是等位基因 110, 的交换仅发生在每个受到影响的染色体 100 的单个点处。多点交叉是当基因材料, 也就是等位基因 110, 的交换仅发生在每个受到影响的染色体 100 的多个点处 (如双点交叉执行双亲中的两个点之间的交换)。一致交叉是当来自双亲的基因被随机打乱。用于 C_i 交叉的 i 的值可以从 1 变化到 n , 其中 n 是个体 120 中的染色体 100 的数量。优选地, 根据本发明的用于 C_i 交叉的 i 的值从 2 到 $n-1$ 。更加优选地, 用于 C_i 交叉的 i 的值是 2。本发明的优选的 C_2 交叉可以包含任意类型的交叉, 包含但不限于单点、双点, 或者一致。优选地, 优选的 C_2 交叉包含单点类型的交叉。

[0061] 图 5 表示在两个个体 120 之间的一种单点、 C_2 交叉。在一种单点 C_2 交叉中,从个体中随机选择两个染色体从而对其进行交叉。然后为两个个体 120 随机选择相同的交叉点 130。染色体 100 上的位于交叉点 130 之后的等位基因 110 在两个个体 120 之间被交换。所得到的个体 120' 被显示在图 5 的底部。恰好有两个染色体经历了交叉。

[0062] 在本发明中使用的另一种改进被称作 C_i 突变。 C_i 突变描述了这样一种突变的发生,这种突变恰好影响一个个体 120 中的 i 个染色体 100。尽管只有 i 个染色体 100 受到 C_i 突变的影响,在每个染色体 100 上可以有多于一个的突变。在单个的染色体 100 上可以发生的突变的数量的范围可以从 1 到 m ,其中 m 是一个染色体 100 中的基因 105 的数量(这由突变的概率所决定)。进一步,如果有多于一个染色体 100 受到突变的影响(如果 i 大于 1),那么每个受到影响的染色体 100 可以具有相等或者不相等的突变的数量。

[0063] 用于 C_i 突变的 i 的值可以从 1 变化到 n ,其中 n 是个体 120 中染色体 100 的数量。优选地,根据本发明的用于 C_i 突变的 i 的值为从 2 到 $n-1$ 。更加优选地,用于 C_i 突变的 i 的值为 2。

[0064] 图 6 描述了 C_2 突变。个体 120 至少具有两个染色体 100 和 100'。在 C_2 突变这种特定的例子中,随机选择两个染色体用于对其进行突变。然后如通常所做的那样,按照突变的概率(在初始化时被定义或者被某种其它的方法所定义)将突变应用于每个被选择的染色体的每个基因。在突变点 140、142 和 144 处的基因 105 的等位基因 110 会被不同的等位基因 110 所替代。得到的突变的染色体 100'' 和 100''' 导致了突变的后代个体 120'。

[0065] 根据本发明在基因算法中使用的另一种改进是对选择要在交配步骤 175 中进行交配的双亲的方法的改进。通常,双亲都是被随机选择的,或者双亲是基于它们的适合性被选择的(如通过在前面所提到过的轮盘选择、比赛选择、等级选择)。在本发明的基因算法中使用的改进会导致一种被称作国王(king)基因算法的基因算法。在国王基因算法中,被选择用于交配的第一双亲总是总体中最适合的个体 120。总体中最适合的个体 120 是通过在算法中使用的适合性的特定的测量标准来被确定的。该双亲被用作第一配偶从而创造下一代的每个成员。被选择用来与第一双亲进行交配的双亲被称作第二双亲,并且该第二双亲是通过一种随机方法来选择的。用于选择第二双亲的方法包含但不限于轮盘选择、比赛选择,或者随机数生成。

[0066] 该改进不同于基本的基因算法,这是因为基本的基因算法通常使用相同类型的方法来选择双亲。例如,通过转盘选择来选择双亲或者通过比赛选择来选择双亲。

[0067] 尽管根据本发明的基因算法包含那些具有上述三种改进中的任意一种改进或者这些改进的组合的基因算法,但本发明优选的基因算法为使用 C_2 突变的国王基因算法和使用 C_2 交叉的国王基因算法。使用 C_2 突变的国王基因算法包含选择总体中最适合的个体作为双亲,接着仅进行 C_2 类型的突变(仅对 2 个染色体 100 进行突变)。因为仅发生突变(交叉的概率为零, $P_c = 0$),所以仅需要出现一对双亲,因而第二双亲就不用被选择。但是,在任意一个染色体 100 上可以发生突变的基因 105 的数量是不受限制的,并且在两个发生突变的染色体 100 上所发生的突变的数量不必是相等的。

[0068] 本发明的第二优选的基因算法为使用 C_2 交叉和 C_2 突变的国王基因算法。该算法包含选择总体中最适合的个体 120 作为第一双亲,随后随机选择第二双亲,并且仅进行 C_2 类型的交叉和突变(仅在两个染色体上进行交叉和突变)。但是,可以发生突变的基因 105

的数量,或者说在任意一个染色体 100 上的交叉点的数量并不限于 1。另外,在两个不同的染色体 100 上的突变或交叉点的数量并不必相同。

[0069] 基因算法在 UGS 网络上的应用

[0070] 本发明的基因算法的一种实际应用包含 UGS 网络的控制和管理。接下来将描述可以通过根据本发明的一种基因算法从而被管理和控制的 UGS 网络的例子。

[0071] 这样一种网络的一个例子包含可以报告目标的分类或标识和目标的方位角的声学传感器。这样一种传感器网络几乎可以具有任意数量的传感器。传感器的数量部分地由被监视的区域、要被执行的任务、传感器的视野领域和范围等决定。通常,将以下这样一些任务目的分配给这种 UGS 网络:对进入监视区域的目标进行探测、跟踪和分类并且使传感器的组合功率消耗最小化(即延长网络的工作寿命)。

[0072] 例如,为了使用方位角数据通过三角关系测量从而准确地对目标进行定位,生成对于目标是最小的位置误差的一组三个传感器的集合为最佳传感器集合。通过使用可以被应用于 UGS 网络的运行的费用标准和限制组合搜索空间的一种有效的最佳策略,作为一个网络进行工作的多个 UGS 可以优化地自我组织和对其自身进行管理从而实现远程区域监视。

[0073] 为了确定用于可以对一种示例性 UGS 网络进行控制的本发明的基因算法的参数,有必要更充分地规定跟踪过程。对于一个 UGS 网络来说,一个想要的属性是能够跟踪目标到任何地方,而不受道路的限制。因此,优选地是具有一个可以实现不受限制的跟踪的 UGS 网络。跟踪是这样一种过程:它通过传感器测量结果来确定在传感器视野内的所有目标的位置。当使用只感测声音、方位角的传感器处理时,为了执行跟踪,每一个目标需要三个传感器。

[0074] 优化的目标是选择 UGS 网络中的传感器的一个集合,该集合可以在以最小的误差完成跟踪过程,同时使费用标准最小化。当可以使用不同的费用标准时,经常被考虑使用的一种通常的标准是每个时刻传感器所使用的总能量。当考虑需要完成多个目的时(即目标探测、跟踪,以及传感器功率使用的最小化),为了获得最佳性能,网络不得不对它的传感器的使用进行优化从而满足这些目的函数中的每一个。

[0075] 本发明的一种基因算法被用来选择传感器的准优化集合从而对目的进行优化。这一问题被视作并不存在唯一解决方案的多目的优化问题。进一步,对于数量按线性规律增加的目标或传感器,可能的解决方案的数量将会导致组合搜索空间按指数规律增加。为了选择可以提供最佳性能的传感器集合,需要为网络目的中的每个目的提供适合的测量标准或费用标准。

[0076] 使用本发明的一种基因算法可以最有效地实现对目的函数的优化。现在将结合图 7 解释本发明的基因算法所基于的结构的一个例子。基因算法总体 125 的每个个体 120 包含多个染色体 100。每个染色体 100 包含构成传感器标识的多个基因 105。被基因算法选择从而在任意给定时刻起作用的所有传感器具有在染色体中被编码的唯一的、二进制编码的标识,也就是基因 105 的等位基因 110。网络目的包含可疑的目标和相关于该目标的所需操作。对于跟踪,有多少被用来实现跟踪的传感器,在个体中就有多少染色体 100。

[0077] 作为一个例子,假设要对 5 个目标进行跟踪,并且跟踪每个目标需要 3 个传感器。另外假设每个染色体 100 包含数量足够多的基因 105 从而具有一个传感器的一个唯一的二

进制标识。在这种情况下,每个个体 120 都将具有代表跟踪 5 个目标所必需的 15 个传感器的 15 个染色体 100。在这 15 个染色体 100 当中,有可能(并且通常代表了最佳解决方案)一个传感器被代表了多于一次。如果一个传感器被代表了多于一次,这意味着一个给定的传感器将被用于跟踪多于一个目标。总体 125 中个体 120 的数量取决于基因算法的特殊的设计。

[0078] 一种使用本发明的基因算法的适合性函数可以处理用户所希望处理的任意数量的变量。可能的变量的例子包含效率、传感器寿命、费用、跟踪误差,以及获得信息的速度。一个示例性适合性函数处理两个目的:使目标位置的准确度最大化(即使位置跟踪误差最小化)和使网络功率消耗最小化。该适合性函数可以被表示为如下形式。

$$[0079] \quad F = - \left(w_1 \sum_{i=1}^n E_i + w_2 \sum_{j=1}^m P_j \right)$$

[0080] 其中 E_i ($i = 1, 2, \dots, n$) 是为第 i 个目标所估计的位置误差; P_j ($j = 1, 2, \dots, m$) 是第 j 个传感器的功率消耗值; n 是目标的数量; m 是所选择的传感器的总数量,并且 w_1 和 w_2 是两个加权常数。 w_1 和 w_2 的值将取决于使误差最小化和使功率消耗最小化的相对重要性。

[0081] 这种用于基因算法和适合性函数 F 的结构可以与根据本发明的基因算法相结合从而创造用来控制和管理一个 UGS 传感器网络的方法。

[0082] 工作实例

[0083] 下面的实例举例说明了本发明的应用和好处,并且这些实例并不对本发明起限制作用。

[0084] 实例 1

[0085] 根据本发明的算法和不根据本发明的算法都被用来对 Rastrigin 的函数进行优化。Rastrigin 的函数由下面的方程给出:

$$[0086] \quad f_4(x_1, \dots, x_{10}) = 200 + \sum (x_i^2 - 10 \cos(2\pi x_i))$$

[0087] Rastrigin 的函数由 10 个独立的变量所确定,并且按照这种形式的 Rastrigin 的函数被认为是大规模多模式的。为了使用基因算法解这个函数,每个独立的变量被编码为基因算法总体中的独立的染色体。在这种情况下每个个体包含 10 个染色体。

[0088] 使用 8 种不同版本的基因算法对该函数进行优化。第一种算法为使用非特定交叉和突变的一种基本的基因算法(表 1 中的 GA)。接下来是也使用交叉和突变但是其中的交叉仅限于 C_2 类型的交叉的基本的基因算法(表 1 中的 GA_C2)。在此之后是仅使用非特定突变的基本的基因算法(表 1 中的 GA 突变)。然后是仅使用 C_2 突变的基本的基因算法(表 1 中的 GA 突变_C2)。接下来是使用非特定突变和交叉的国王基因算法(表 1 中的国王 GA)。接下来是仅使用非特定突变和 C_2 交叉的国王基因算法(表 1 中的国王 GA_C2)。仅使用非特定突变的国王基因算法(表 1 中的国王突变)。最后是仅使用 C_2 突变的国王基因算法(表 1 中的国王突变_C2)。

[0089] 该表给出了用于所检查的每种不同的基因算法的交叉的概率 P_c , 以及突变的概率 P_m 。总体大小以及被重复的代的数量对于所检查的不同的算法来说是保持一致的,并且分别是 100 和 450。最佳次数代表函数的最佳值被确定时运行过的次数。每种算法总共被运

行 30 次。最佳次数和运行的总次数被用来计算各种算法的有效性,该有效性是趋同到全局优化的运行所占的百分比。

[0090] 表 1:不同的基因算法在优化 Rastringin 函数方面的性能

[0091]

| 方法 | 交叉的概率 P_c | 突变的概率 P_m | 总体大小 P_s | 基因的数量 | 最佳次数 | 运行次数 | 有效性 |
|----------|----------------|----------------|---------------|-------|------|------|------|
| GA | 0.9 | 0.01 | 100 | 450 | 6 | 30 | 0.20 |
| GA-C2 | 0.9 | 0.0625 | 100 | 450 | 11 | 30 | 0.37 |
| GA 突变 | 0 | 0.01 | 100 | 450 | 1 | 30 | 0.03 |
| GA 突变-C2 | 0 | 0.0625 | 100 | 450 | 17 | 30 | 0.57 |
| 国王 GA | 0.9 | 0.01 | 100 | 450 | 18 | 30 | 0.60 |
| 国王 | 0.9 | 0.0625 | 100 | 450 | 29 | 30 | 0.97 |
| GA-C2 | | | | | | | |
| 国王突变 | 0 | 0.01 | 100 | 450 | 2 | 30 | 0.07 |
| 国王突变-C2 | 0 | 0.0625 | 100 | 450 | 30 | 30 | 1.00 |

[0092] 仅发生 C_2 突变的国王基因算法 (国王突变 C_2) 给出了所有被研究的基因算法的最佳结果。当与不使用这些本发明的改进的基本的基因算法进行比较的时候,有效性增加了 5 倍。

[0093] 实例 2

[0094] 将来自上述实例 1 的最佳表现的算法与在 K. Deb, S. Agrawal 所写的“Understanding Interactions Among Genetic Algorithm Parameters (理解基因算法参数之间的相互作用)” (Foundations of Genetic Algorithm 5 (基因算法基础 5), W. Banzhaf, C. Reeves (eds.), Morgan Kaufmann 出版有限公司, San Francisco, CA, 第 265-186 页, 1999 (“Deb”)) 中被测试的最佳基因算法进行比较。

[0095] 如上面给出的,为了优化的 Rastringin 的函数,对 Deb 的最佳基因算法进行测试。与用于 Deb 中基因算法的总体大小 1000 相比,仅使用 C_2 突变的国王基因算法的总体大小对于两次运行都为 10。来自该参考的基因算法仅在具有大的总体的时候才运行良好,并且对来自参考的那些被使用的基因算法来说,总体大小为 1000 是最佳的。

[0096] 下面的表 2 给出了使用根据本发明的基因算法和来自 Deb 的最佳基因算法的结果。该表给出了用于所检查的每种不同的基因算法的突变的概率 P_m , 以及交叉的概率 P_c 。该表还给出了总体大小以及被重复的代的数量,并且可以从中看出它们对于所检查的不同的算法来说并不是保持一致的。重要的因素是由每种算法所进行的适合性函数评价的次数。

该值是通过将总体大小与代的数量相乘而得到的。由于每个这样的计算所花费的名义上的时间的缘故,该值是重要的。对适合性函数所必须进行的评价的次数越少,就可以越快实现对函数的优化。

[0097] 该最佳次数代表获得函数的最佳值时运行过的次数。运行的次数对于根据本发明的基因算法和来自 Deb 的那些基因算法来说也是不同的。然后,基于优化运行的次数来计算有效性。该表还显示了函数必须被评价的次数(函数评价次数),该次数被用来计算根据本发明的两种基因算法相对于来自 Deb 的最佳算法所节省的时间。

[0098] 表 2:国王突变 C2 和 Deb 算法在优化 Rastringin 函数方面的性能

[0099]

| 方法 | P ₀ | P _n | 总体大小 | 基因数目 | 优化次数 | 运行次数 | 有效性 | 函数评价的次数 | 节省时间 |
|--------------|----------------|----------------|------|------|------|------|------|---------|-------|
| 国王突变 C2 | 0 | 0.1 | 10 | 1000 | 24 | 30 | 0.80 | 10000 | 64.2% |
| 国王突变 C2 | 0 | 0.1 | 10 | 2000 | 30 | 30 | 1.00 | 20000 | 28.3% |
| 来自 Deb 的最佳结果 | 0.9 | 0 | 1000 | 45 | 45 | 50 | 0.90 | 27900 | 0.00% |

[0100] 实例 3

[0101] 在该例中,将本发明的基因算法与用于一个“欺骗函数”的基本的基因算法进行比较。在该例中被优化的函数为单位函数。该单位函数为这样一种函数:它的值仅取决于它所作用的字符串中的 1 和 0 的数量。单位函数 u 计算一个字符串中的 1 的数量。在该例中被优化的欺骗函数就具有下面的数学表达形式:

$$[0102] \quad f_s = \sum_{i=1}^{10} g(u_i)$$

[0103] 其中 u 是单位函数。

[0104] 在下面的表 3 中给出了单位函数 u 的值从 0 到 4 时函数 $g(u)$ 的值。

[0105] 表 3:对于 u 的值从 0 到 4 的情况的 $g(u)$ 的值

[0106]

| | | | | | |
|-------------|----------|----------|----------|----------|----------|
| u | 0 | 1 | 2 | 3 | 4 |
| g(u) | 3 | 2 | 1 | 0 | 4 |

[0107] 所以,对于一个 4 位字符串,下面的表 4 给出了 $g(u)$ 的结果:

[0108] 表 4:对于 4 位字符串的 $g(u)$ 的值

[0109]

| 字符串 (4 位) | u | g(u) |
|-----------|---|------|
| 0000 | 0 | 3 |
| 0001 | 1 | 2 |
| 0010 | 1 | 2 |
| 0100 | 1 | 2 |
| 1000 | 1 | 2 |
| 0011 | 2 | 1 |
| 0101 | 2 | 1 |
| 0110 | 2 | 1 |
| 1010 | 2 | 1 |
| 1100 | 2 | 1 |
| 0111 | 3 | 0 |
| 1011 | 3 | 0 |
| 1101 | 3 | 0 |
| 1110 | 3 | 0 |
| 1111 | 4 | 4 |

[0110] f_s 是一种难解的欺骗函数,这是因为相应于欺骗吸引子 (attractor) (全 0 字符串) 的低阶构建块比相应于全局吸引子 (全 1 字符串) 的低阶构建块要好。

[0111] 被检查的基因算法包含与在上述实例 1 中被检查的 8 种变化相同的变化,并且包含以下的变化。第一种变化是使用非特定交叉和突变的基本的基因算法 (下面表 5 中的 GA)。接下来是也使用交叉和突变但是其中的交叉仅限于 C_2 类型的交叉的基本的基因算法 (表 5 中的 GA- C_2)。在此之后是仅使用非特定突变的基本的基因算法 (表 5 中的 GA 突变)。然后是仅使用 C_2 突变的基本的基因算法 (表 5 中的 GA 突变- C_2)。接下来是使用非特定突变和交叉的国王基因算法 (表 5 中的国王 GA)。接下来是仅使用非特定突变和 C_2 交叉的国王基因算法 (表 5 中的国王 GA- C_2)。然后检查的是仅使用非特定突变的国王基因算法 (表 5 中的国王突变)。最后是仅使用 C_2 突变的国王基因算法 (表 5 中的国王突变- C_2)。

[0112] 下面的表 5 给出了这些比较的结果。该表给出了用于所检查的每种不同的基因算

法的交叉的概率 P_c , 以及突变的概率 P_m 。总体大小以及经历的代的数目对于所检查的不同的方法来说是保持一致的, 并且分别是 100 和 450。最佳次数代表函数的最佳值被确定时运行过的次数。每种算法总共被运行 30 次。最佳次数和运行的总次数被用来计算各种算法的有效性。

[0113] 表 5: 本发明的不同基因算法改进在欺骗函数优化方面的性能

[0114]

| 方法 | 交叉的概率 P_c | 突变的概率 P_m | 总体大小 P_s | 基因的数量 | 最佳次数 | 运行次数 | 有效性 |
|----------|----------------|----------------|---------------|-------|------|------|------|
| GA | 0.9 | 0.025 | 100 | 150 | 0 | 30 | 0.00 |
| GA-C2 | 0.9 | 0.25 | 100 | 150 | 1 | 30 | 0.03 |
| GA 突变 | 0 | 0.025 | 100 | 150 | 0 | 30 | 0.00 |
| GA 突变-C2 | 0 | 0.25 | 100 | 150 | 6 | 30 | 0.20 |
| 国王 GA | 0.9 | 0.025 | 100 | 150 | 0 | 30 | 0.00 |
| 国王 GA-C2 | 0.9 | 0.25 | 100 | 150 | 22 | 30 | 0.73 |
| 国王突变 | 0 | 0.025 | 100 | 150 | 0 | 30 | 0.00 |
| 国王突变-C2 | 0 | 0.25 | 100 | 150 | 29 | 30 | 0.97 |

[0115] 与基本的 GA 的结果为 0.0 的有效性相比, 国王突变 C2 实现了达到 0.97 的非常高的有效性。

[0116] 实例 4

[0117] 本发明的基因算法与基本的基因算法相比较, 用于对用于跟踪 7 个目标的传感器测试函数进行优化。

[0118] 在该实例中被模拟的传感器网络包含可以报告目标的分类或识别和方位角的声学传感器。该被模拟的传感器网络具有 181 个传感器, 这 181 个传感器每个都具有 360° 的 FOV (视野), 其中半径为 4km, 并且这 181 个传感器被随机地分布在面积为 625km^2 的监视区域上。

[0119] 该网络的任务目的是对进入监视区域的目标进行探测、跟踪和分类并且使传感器的联合功率消耗最小化 (即延长网络的工作寿命)。例如, 为了使用方位角数据通过三角关系测量从而准确地对目标进行定位, 在最低的联合功率消耗下生成对于目标来说是最小的位置误差的一组三个传感器的集合为最佳传感器集合。为了确定可以被优化的目的函数, 有必要对这两个因素进行特殊地加权。

[0120] 因为这 7 个目标中的每个目标都需要找到 3 个传感器,所以基因算法中的每个个体包含 $7*3 = 21$ 个染色体。每个染色体包含一个传感器的标识号。被使用的基因算法与在图 8 中描述的基因算法相似。

[0121] 用于这种基因算法结构的适合性函数处理两个目的:使目标位置的准确度最大化(即,使位置跟踪误差最小化)和使网络功率消耗最小化。该适合性函数可以被表达如下。

$$[0122] \quad F = - \left(w_1 \sum_{i=1}^n E_i + w_2 \sum_{j=1}^m P_j \right)$$

[0123] 其中 E_i ($i = 1, 2, \dots, n$) 是为第 i 个目标所估计的位置误差; P_j ($j = 1, 2, \dots, m$) 是第 j 个传感器的功率消耗值; n 是目标的数量; m 是所选择的传感器的总数量,并且 w_1 和 w_2 是两个加权常数。 w_1 和 w_2 的值将取决于使误差最小化和使功率消耗最小化的相对重要性。

[0124] 然后使用模拟的声学传感器测量数据对基因算法进行评价。该模拟的数据包含传感器位置、方位角测量和来自每个传感器的目标识别数据。对属于被跟踪的交通工具的类的 7 个目标的运动轨迹进行模拟。这些目标位于相同的邻近范围内,这意味着最佳传感器选择将会是这样一种选择:其中特定的传感器被共享。

[0125] 表 6:用于 7 个目标的不同基因算法在优化适合性函数方面的性能

[0126]

| | | | | | | | | | |
|----|--|-------|---|----|----|---|---|----|-----|
| 方法 | | P_m | 总 | 基因 | 基因 | 最 | 运 | 有效 | 平均最 |
|----|--|-------|---|----|----|---|---|----|-----|

[0127]

| | P_c | | 体 大 小 n | 权重 变化 数 | 运行 的次 数 | 佳 次 数 | 行 次 数 | 性 | 佳适合 性 |
|-------------------|-------|------|---------------|---------------|---------------|-------------|-------------|------|----------|
| GA | 0.9 | 0.01 | 10 | 2000 | 4492 | 3 | 20 | 0.15 | -773.4 |
| GA-C ₂ | 0.9 | 0.1 | 10 | 2000 | 3608 | 8 | 20 | 0.40 | -714.2 |
| GA 突 变 | 0 | 0.01 | 10 | 2000 | 4655 | 7 | 20 | 0.35 | -679.9 |
| GA 突 变 C2 | 0 | 0.1 | 10 | 2000 | 3524 | 8 | 20 | 0.40 | -660.7 |
| 国王 GA | 0.9 | 0.01 | 10 | 2000 | 4138 | 6 | 20 | 0.30 | -675.4 |
| 国王 GA C2 | 0.9 | 0.1 | 10 | 2000 | 3764 | 14 | 20 | 0.70 | -576.9 |
| 国王 突变 | 0 | 0.01 | 10 | 2000 | 3270 | 9 | 20 | 0.45 | -647.2 |
| 国王 突变 C2 | 0 | 0.1 | 10 | 2000 | 3299 | 14 | 20 | 0.70 | -599.0 |

[0128] 图 9 是描述所使用的不同算法的平均最佳适合性的图。可以看出,无论所使用的是哪种基因算法,那些仅使用 C₂ 交叉或突变的基因算法的工作情况总是更好一些。

[0129] 图 10 比较了在表 6 中被检查的 5 种不同的基因算法的有效性和所必需的时间。在图 10 中被描述的这些方法包含以下一些基因算法:没有进行试验并且总体大小为 50 的基本的基因算法、经过试验后的基本的基因算法(较小的总体大小会产生更好的有效性)、仅使用突变的基本的基因算法、仅使用突变的国王基因算法,以及仅使用 C₂ 类型突变的国王基因突变。

[0130] 图 11 描述了对于在上述图 10 中描述的相同的 5 种基因算法变化,其百分比改进随时间发生变化的情况。

[0131] 以上的详述、实例和数据提供了对本发明的组成的制造和使用的完整的描述。因为可以在不偏离本发明的精神和范围的情况下实现本发明的多种实施方案,所以本发明属于此后所附的权利要求。

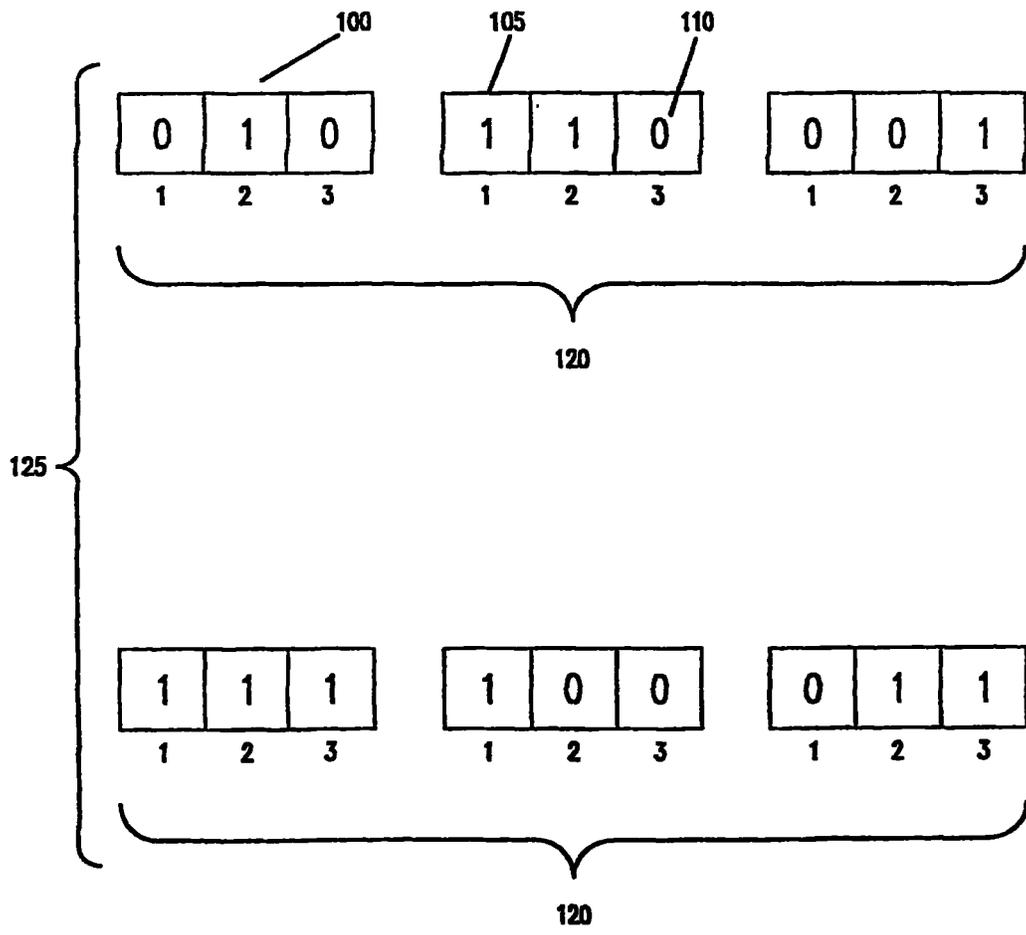


图 1

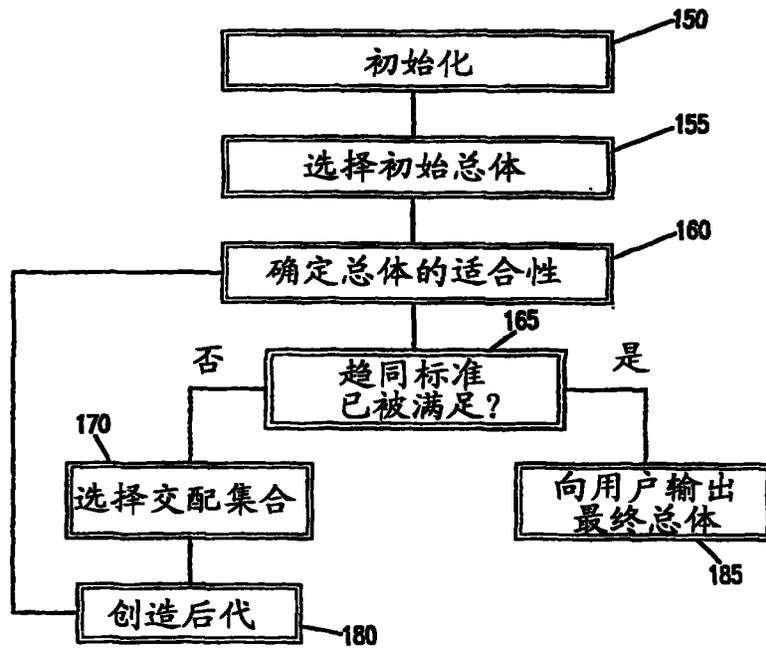


图 2

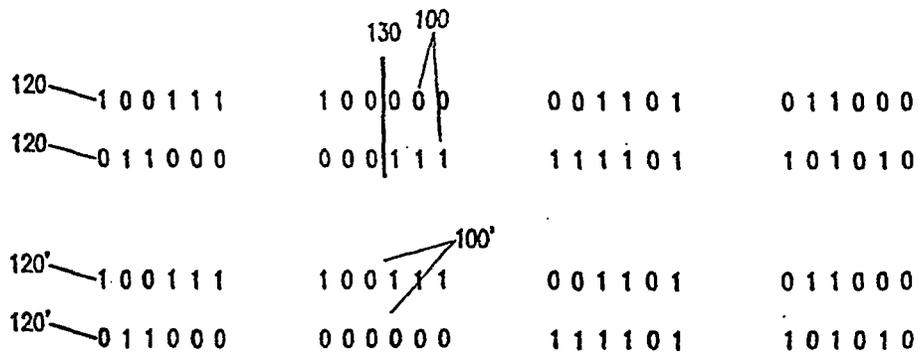


图 3A

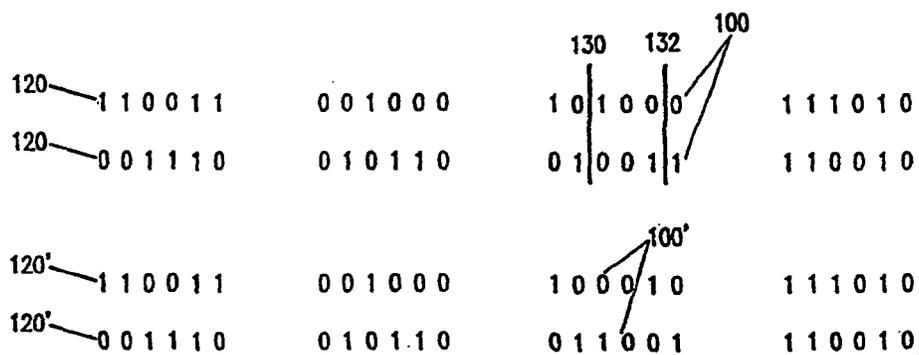


图 3B

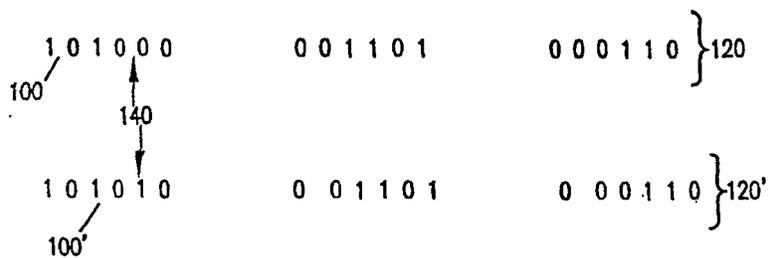


图 4A

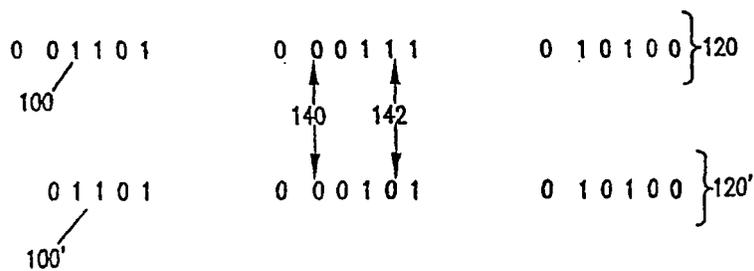


图 4B

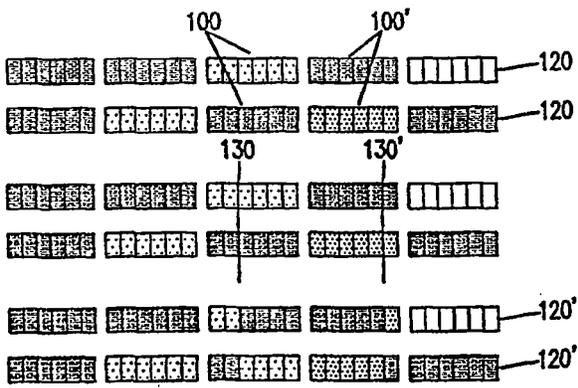


图 5

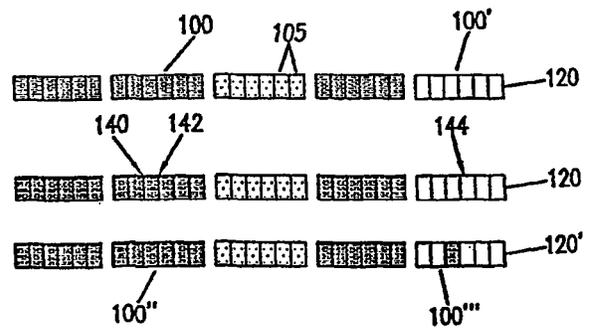


图 6

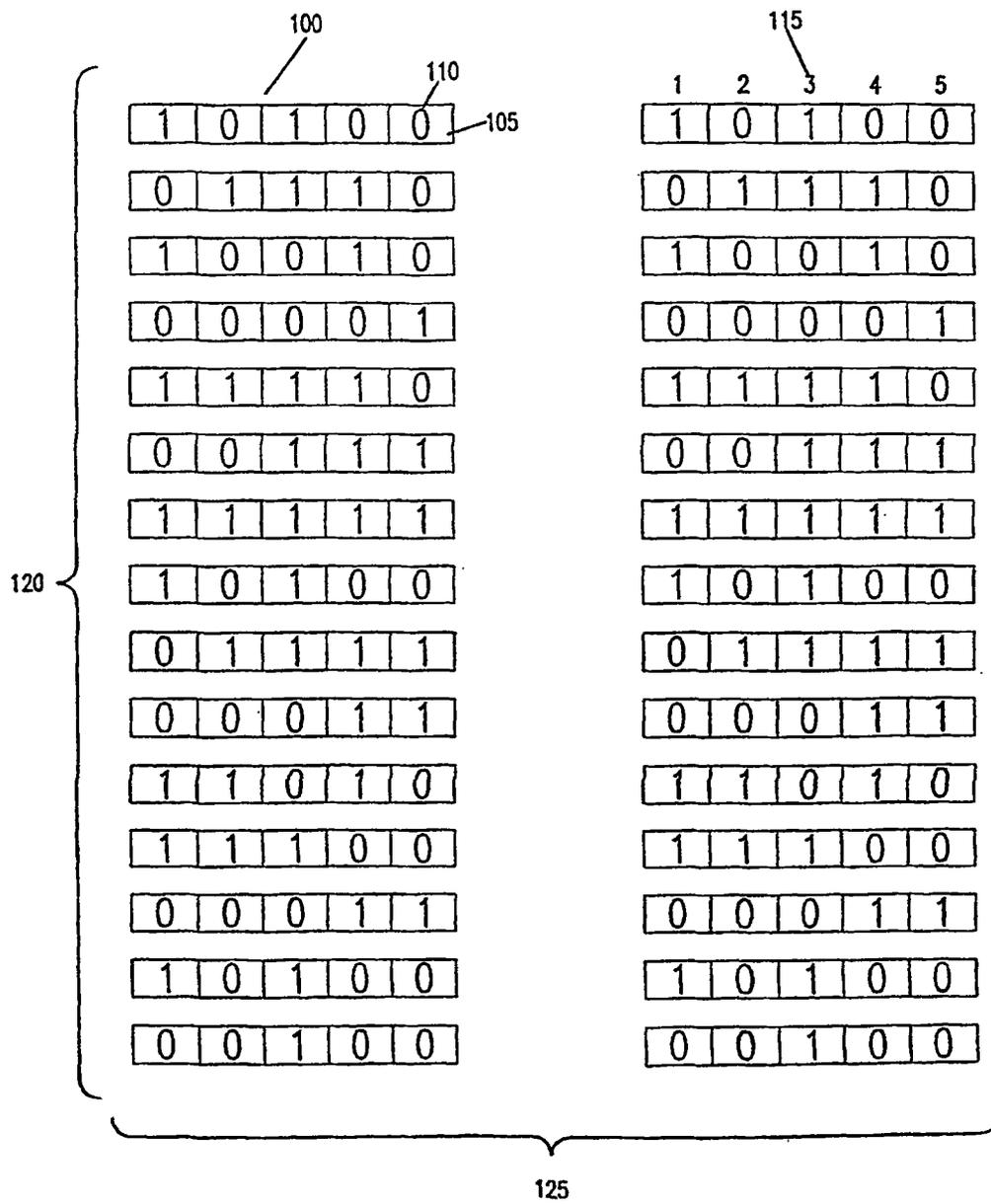


图 7

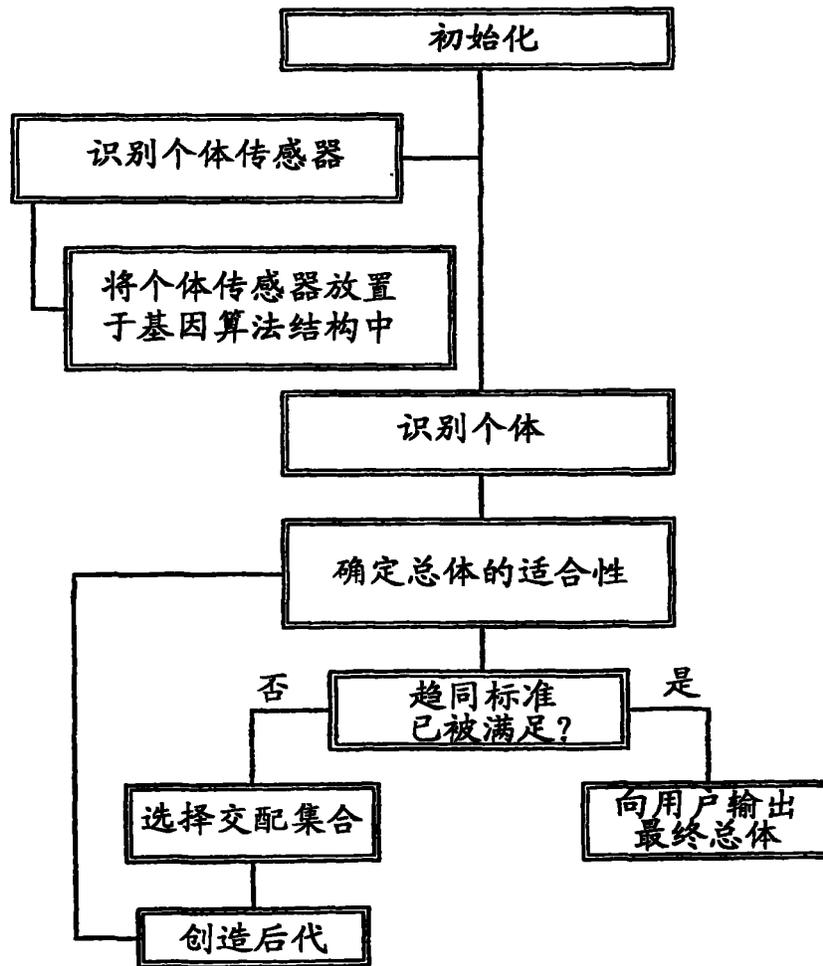


图 8

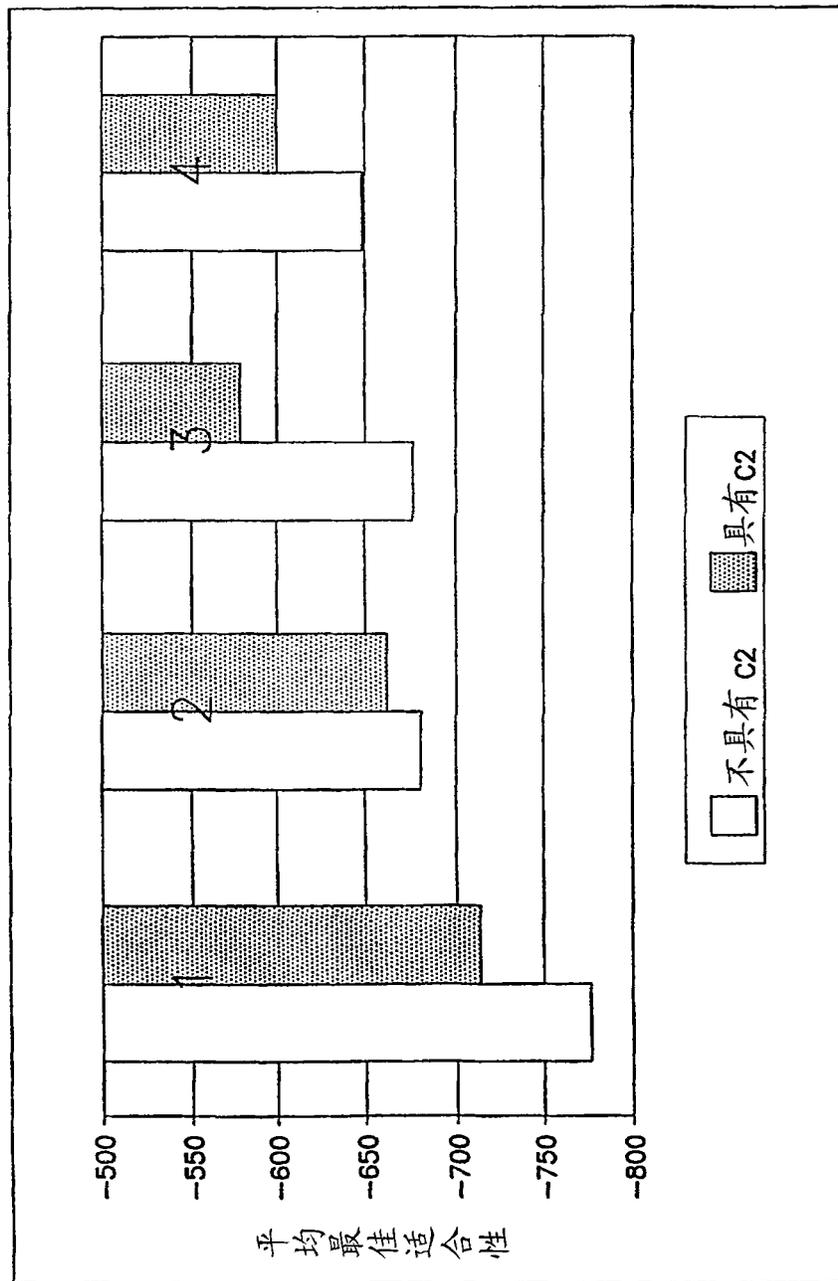


图 9

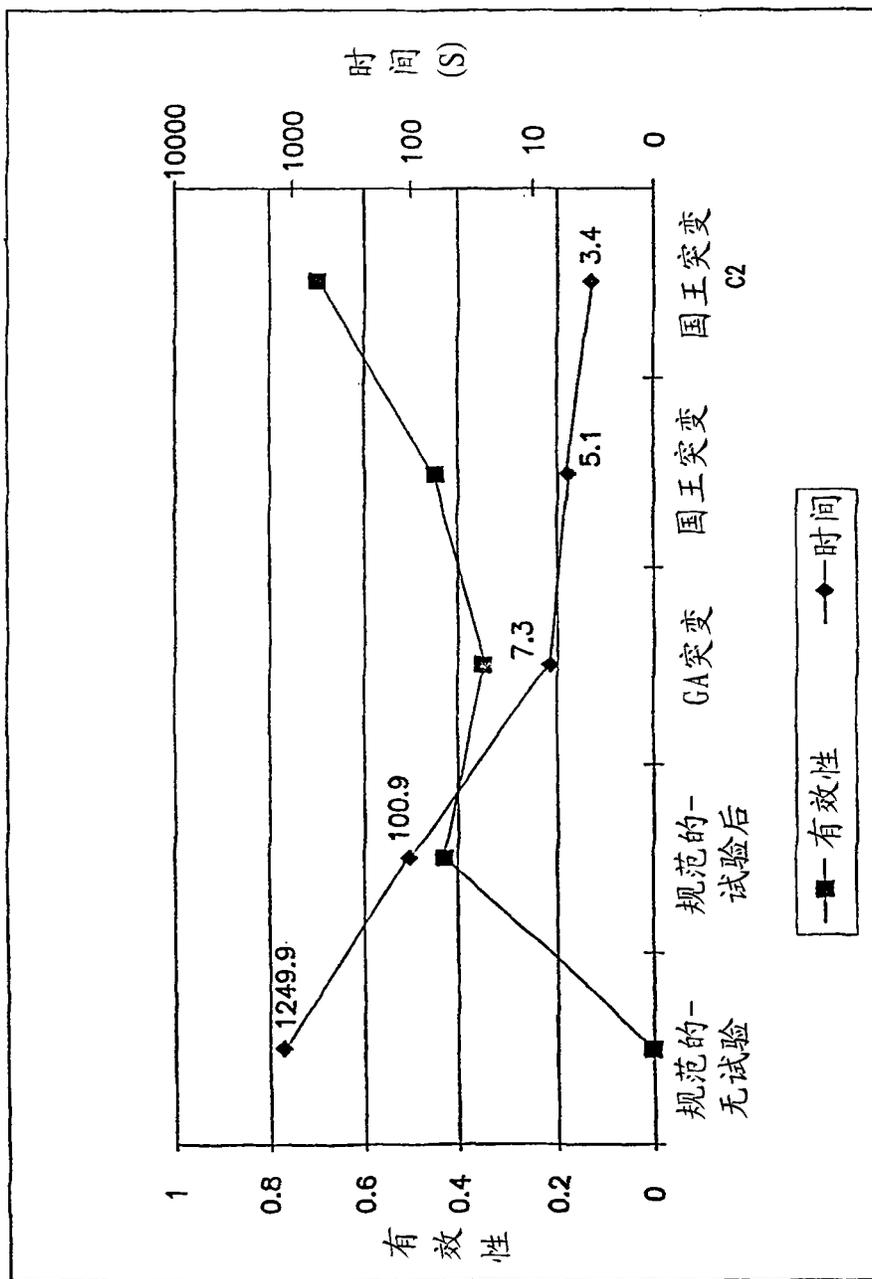


图 10

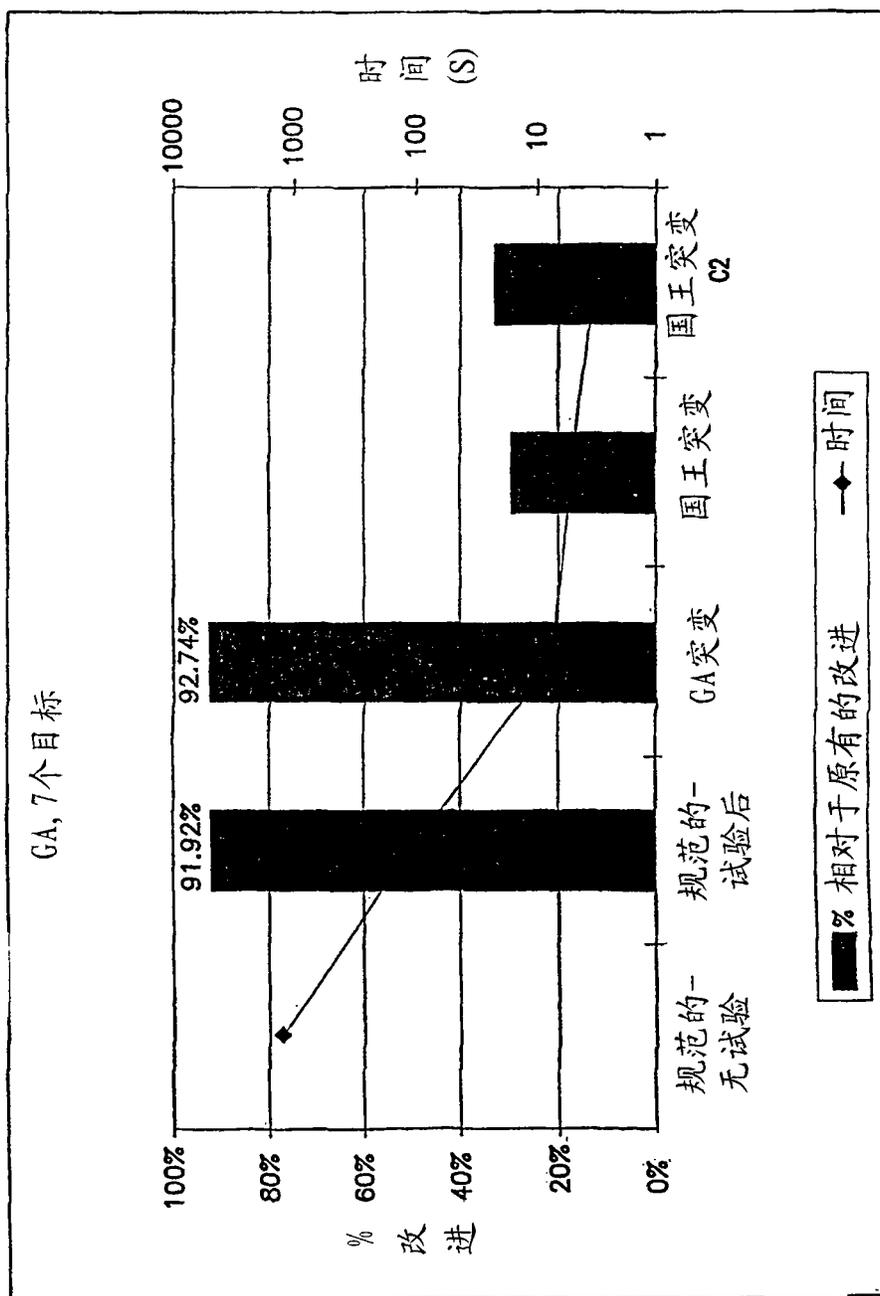


图 11