



(12) 发明专利

(10) 授权公告号 CN 110062979 B

(45) 授权公告日 2022. 07. 08

(21) 申请号 201780075856.7  
 (22) 申请日 2017.01.09  
 (65) 同一申请的已公布的文献号  
 申请公布号 CN 110062979 A  
 (43) 申请公布日 2019.07.26  
 (30) 优先权数据  
 15/400488 2017.01.06 US  
 (85) PCT国际申请进入国家阶段日  
 2019.06.06  
 (86) PCT国际申请的申请数据  
 PCT/US2017/012673 2017.01.09  
 (87) PCT国际申请的公布数据  
 W02018/128626 EN 2018.07.12  
 (73) 专利权人 维斯柯航空有限公司  
 地址 美国加利福尼亚州  
 (72) 发明人 L.R.洪 T.P.穆尼斯 P.K.赫林

(74) 专利代理机构 上海专利商标事务所有限公  
 司 31100  
 专利代理师 周全 钱慰民

(51) Int.Cl.  
 H01M 10/42 (2006.01)  
 H01M 10/48 (2006.01)  
 H01M 10/04 (2006.01)  
 H01M 6/50 (2006.01)

(56) 对比文件  
 TW 201621532 A, 2016.06.16  
 CN 103956513 A, 2014.07.30  
 US 2013214730 A1, 2013.08.22  
 US 2010040939 A1, 2010.02.18  
 US 5780992 A, 1998.07.14  
 CN 101814632 A, 2010.08.25  
 US 2009130541 A1, 2009.05.21  
 JP 2012252926 A, 2012.12.20  
 JP 2015187915 A, 2015.10.29

审查员 薛召

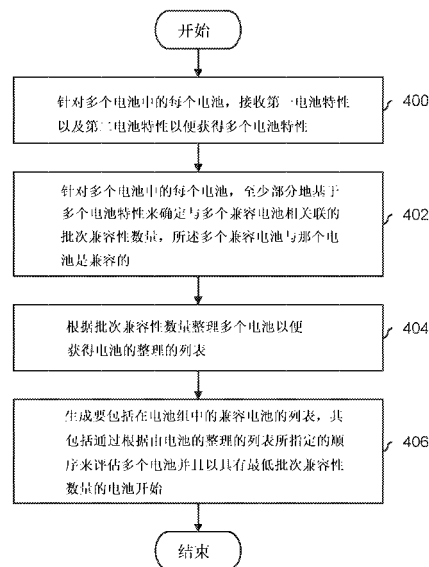
权利要求书4页 说明书15页 附图13页

(54) 发明名称

在电池组装配期间跨多个特性的电池匹配

(57) 摘要

针对多个电池中的每个电池,接收第一电池特性以及第二电池特性以便获得多个电池特性。针对多个电池中的每个电池,至少部分地基于多个电池特性来确定与多个兼容电池相关联的批次兼容性数量,所述多个兼容电池与那个电池是兼容的。根据批次兼容性数量整理多个电池以便获得电池的整理的列表。生成要包括在电池组中的兼容电池的列表,其包括通过根据由电池的整理的列表所指定的顺序来评估多个电池并且以具有最低批次兼容性数量的电池开始。



1. 一种用于电池匹配的系统,包括:  
处理器;以及  
存储器,所述存储器与所述处理器耦合,其中所述存储器配置成给所述处理器提供指令,当所述指令被执行时使得所述处理器进行以下操作:  
针对多个电池中的每个电池,接收第一电池特性和第二电池特性以便获得多个电池特性;  
针对所述多个电池中的每个电池,至少部分地基于所述多个电池特性来确定指示兼容电池数量的批次兼容性数量,所述兼容电池与那个电池是兼容的;  
根据所述批次兼容性数量对所述多个电池进行排序以便获得电池的排序的列表;以及  
生成要包括在电池组中的兼容电池的列表,其包括通过根据由所述电池的排序的列表所指定的顺序来评估所述多个电池,所述评估以具有最低批次兼容性数量的电池开始并且包括:将具有最低批次兼容性数量的电池包括在所述要包括在电池组中的兼容电池的列表中,将正被评估的电池与所述要包括在电池组中的兼容电池的列表中的每一个电池进行比较,以确定它们是否相互兼容,其中所述排序的列表被持续更新,以从所述排序的列表中排除已被选择要包含在兼容电池列表中的电池,所述已被选择要包含在兼容电池列表中的电池是与所述要包括在电池组中的兼容电池的列表中的每一个电池均兼容的电池。
2. 根据权利要求1所述的用于电池匹配的系统,其中所述第一电池特性包括以下中的一者:容量、开路电压、电阻或自放电速率。
3. 根据权利要求1所述的用于电池匹配的系统,其中确定所述批次兼容性数量包括生成电池兼容性矩阵,所述电池兼容性矩阵针对所述多个电池指示给定电池对是否兼容。
4. 根据权利要求1所述的用于电池匹配的系统,其中确定所述批次兼容性数量包括:  
针对所述第一电池特性确定所述多个电池中的第一电池与所述多个电池中的第二电池之间的第一差;  
针对所述第二电池特性确定所述第一电池与所述第二电池之间的第二差;以及  
在所述第一差不超过与所述第一电池特性相关联的第一容差以及所述第二差不超过与所述第二电池特性相关联的第二容差的情况下,宣布所述第一电池与所述第二电池是兼容的。
5. 根据权利要求1所述的用于电池匹配的系统,其中对所述多个电池进行排序包括:  
针对具有相同批次兼容性数量的多个平局电池中的每个平局电池确定兼容电池的集合,以便获得兼容电池的多个集合;以及  
至少部分地基于与所述兼容电池的多个集合相关联的唯一性以及与所述兼容电池的多个集合相关联的一个或多个批次兼容性数量来打破所述多个平局电池之间的平局,其中,打破所述多个平局电池之间的所述平局包括:  
通过移除相同项来使兼容电池的多个集合全局地唯一化,以便获得兼容电池的多个唯一化的集合;  
针对所述多个唯一化的集合中的每个非空唯一化的集合确定最低批次兼容性数量;以及  
至少部分地基于每个非空唯一化的集合的所述最低批次兼容性数量来对所述多个平局电池进行排序,其中当打破所述平局时任何空的唯一化的集合被最后排列。

6. 根据权利要求1所述的用于电池匹配的系统,其中:

对所述多个电池进行排序包括:

针对具有相同批次兼容性数量的多个平局电池中的每个平局电池确定兼容电池的集合,以便获得兼容电池的多个集合;以及

至少部分地基于与所述兼容电池的多个集合相关联的唯一性以及与所述兼容电池的多个集合相关联的一个或多个批次兼容性数量来打破所述多个平局电池之间的平局;以及

其中,打破所述多个平局电池之间的所述平局包括:

至少部分地基于与所述兼容电池的多个集合相关联的所述批次兼容性数量来对所述兼容电池的多个集合进行排序以获得兼容电池的多个排序的列表;

从兼容电池的不同排序的列表中选择用来比较的两个或更多兼容电池,其中所选择的兼容电池在兼容电池的其相应的按上升顺序排序的列表中的相同级别下;以及

在与用来比较的所选择的兼容电池相关联的最小批次兼容性数量是唯一的情况下,选择与所述最小且唯一的批次兼容性数量相关联的平局电池为所述多个平局电池的决胜局获胜者。

7. 一种用于电池匹配的方法,包括:

针对多个电池中的每个电池接收第一电池特性以及第二电池特性以便获得多个电池特性;

针对所述多个电池中的每个电池,至少部分地基于所述多个电池特性来确定指示兼容电池数量的批次兼容性数量,所述兼容电池与那个电池是兼容的;

根据所述批次兼容性数量对所述多个电池进行排序以便获得电池的排序的列表;以及

生成要包括在电池组中的兼容电池的列表,其包括通过根据由所述电池的排序的列表所指定的顺序来评估所述多个电池,所述评估以具有最低批次兼容性数量的电池开始并且包括:将具有最低批次兼容性数量的电池包括在所述要包括在电池组中的兼容电池的列表中,将正被评估的电池与所述要包括在电池组中的兼容电池的列表中的每一个电池进行比较,以确定它们是否相互兼容,其中所述排序的列表被持续更新,以从所述排序的列表中排除已被选择要包含在兼容电池列表中的电池,所述已被选择要包含在兼容电池列表中的电池是与所述要包括在电池组中的兼容电池的列表中的每一个电池均兼容的电池。

8. 根据权利要求7所述的用于电池匹配的方法,其中所述第一电池特性包括以下中的一者:容量、开路电压、电阻或自放电速率。

9. 根据权利要求7所述的用于电池匹配的方法,其中确定所述批次兼容性数量包括生成电池兼容性矩阵,所述电池兼容性矩阵针对所述多个电池指示给定电池对是否兼容。

10. 根据权利要求7所述的用于电池匹配的方法,其中确定所述批次兼容性数量包括:

针对所述第一电池特性确定所述多个电池中的第一电池与所述多个电池中的第二电池之间的第一差;

针对所述第二电池特性确定所述第一电池与所述第二电池之间的第二差;以及

在所述第一差不超过与所述第一电池特性相关联的第一容差以及所述第二差不超过与所述第二电池特性相关联的第二容差的情况下,宣布所述第一电池与所述第二电池是兼容的。

11. 根据权利要求7所述的用于电池匹配的方法,其中对所述多个电池进行排序包括:

针对具有相同批次兼容性数量的多个平局电池中的每个平局电池确定兼容电池的集合,以便获得兼容电池的多个集合;以及

至少部分地基于与所述兼容电池的多个集合相关联的唯一性以及与所述兼容电池的多个集合相关联的一个或多个批次兼容性数量来打破所述多个平局电池之间的平局,

打破所述多个平局电池之间的所述平局包括:

至少部分地基于与所述兼容电池的多个集合相关联的所述批次兼容性数量来对所述兼容电池的多个集合进行排序以获得兼容电池的多个排序的列表;

从兼容电池的不同排序的列表中选择用来比较的两个或更多兼容电池,其中所选择的兼容电池在兼容电池的其相应的按上升顺序排序的列表中的相同级别下;以及

在与用来比较的所选择的兼容电池相关联的最小批次兼容性数量是唯一的情况下,选择与所述最小且唯一的批次兼容性数量相关联的平局电池为所述多个平局电池的决胜局获胜者。

12. 一种包含用于电池匹配的计算机程序产品的存储介质,所述存储介质是非暂时性计算机可读存储介质,所述计算机程序产品包括用于以下操作的计算机指令:

针对多个电池中的每个电池接收第一电池特性以及第二电池特性以便获得多个电池特性;

针对所述多个电池中的每个电池,至少部分地基于所述多个电池特性来确定指示兼容电池数量的批次兼容性数量,所述兼容电池与那个电池是兼容的;

根据所述批次兼容性数量对所述多个电池进行排序以便获得电池的排序的列表;以及生成要包括在电池组中的兼容电池的列表,其包括通过根据由所述电池的排序的列表所指定的顺序来评估所述多个电池,所述评估以具有最低批次兼容性数量的电池开始并且包括:将具有最低批次兼容性数量的电池包括在所述要包括在电池组中的兼容电池的列表中,将正被评估的电池与所述要包括在电池组中的兼容电池的列表中的每一个电池进行比较,以确定它们是否相互兼容,其中所述排序的列表被持续更新,以从所述排序的列表中排除已被选择要包含在兼容电池列表中的电池,所述已被选择要包含在兼容电池列表中的电池是与所述要包括在电池组中的兼容电池的列表中的每一个电池均兼容的电池。

13. 根据权利要求12所述的包含用于电池匹配的计算机程序产品的存储介质,其中:

用于对所述多个电池进行排序的所述计算机指令包括用于以下操作的计算机指令:

针对具有相同批次兼容性数量的多个平局电池中的每个平局电池确定兼容电池的集合,以便获得兼容电池的多个集合;以及

至少部分地基于与所述兼容电池的多个集合相关联的唯一性以及与所述兼容电池的多个集合相关联的一个或多个批次兼容性数量来打破所述多个平局电池之间的平局;以及

用于打破所述多个平局电池之间的所述平局的所述计算机指令包括用于以下操作的计算机指令:

至少部分地基于与所述兼容电池的多个集合相关联的所述批次兼容性数量来对所述兼容电池的多个集合进行排序以获得兼容电池的多个排序的列表;

从兼容电池的不同排序的列表中选择用来比较的两个或更多兼容电池,其中所选择的兼容电池在兼容电池的其相应的按上升顺序排序的列表中的相同级别下;以及

在与用来比较的所选择的兼容电池相关联的最小批次兼容性数量是唯一的情况下,选

择与所述最小且唯一的批次兼容性数量相关联的平局电池为所述多个平局电池的决胜局获胜者。

## 在电池组装配期间跨多个特性的电池匹配

### 背景技术

[0001] 一些类型的电池组包括多个电池,并且在这样的电池组的装配期间,要包括在特定电池组的电池是经选择的。尽管对其执行电池组匹配的电池的池(pool of cell)可能来自相同批次或制造系列(manufacturing run),但电池的特性(例如,电力特性)中通常将存在一些变化。当特定电池组中的所有电池匹配时,电池组更好地运行。尽管存在用于根据单个特性匹配电池的技术,但不存在用来跨多个电池特性匹配电池的技术。这不是简单的问题,这是因为电池特性常常是不相关的,使得即使两个电池关于一个电池特性是兼容的,以下情况也不一定是适用的:电池关于另一电池特性将也是兼容的。用来跨多个电池特性匹配电池的新技术将是期望的,这是由于它可以降低被丢弃的剩余电池的数量。

### 附图说明

[0002] 在以下详细描述和附图中公开了发明的各种实施例。

[0003] 图1是图示包括多个电池的电池组的实施例的示图。

[0004] 图2是图示用于匹配电池的电池特性的实施例的表格。

[0005] 图3是图示电池特性的分布的实施例的示图。

[0006] 图4是图示用来跨多个电池特性执行电池匹配的过程的实施例的流程图。

[0007] 图5是图示电池兼容性矩阵的实施例的示图。

[0008] 图6是图示用于成对电池兼容性测试的过程的实施例的流程图。

[0009] 图7是图示电池的整理的列表的实施例的示图。

[0010] 图8是图示根据由电池的整理的列表所指定的顺序来评估的电池的实施例的示图。

[0011] 图9是图示决胜局(tiebreaker)过程的实施例的流程图。

[0012] 图10是图示决胜局技术的实施例的示图,所述决胜局技术首先全局地使与平局电池(tiecell)兼容的电池唯一化,并且然后检查唯一化的兼容电池的批次兼容性数量。

[0013] 图11是图示决胜局过程的实施例的流程图,所述决胜局过程首先全局地使与平局电池兼容的电池唯一化,并且然后检查唯一化的兼容电池的批次兼容性数量。

[0014] 图12是图示决胜局技术的实施例的示图,其中与平局电池兼容的电池基于批次兼容性数量被整理,并且然后在逐级别的基础上被评估。

[0015] 图13是图示决胜局过程的实施例的流程图,其中与平局电池兼容的电池基于批次兼容性数量被整理,并且然后在逐级别的基础上针对唯一性被评估。

### 具体实施方式

[0016] 发明可以以多种方式实现,所述多种方式包括作为过程、装置、系统、物质的组成、被包含在计算机可读存储介质上的计算机程序产品和/或处理器,诸如配置成执行存储在耦合到处理器的存储器上和/或由耦合到处理器的存储器提供的指令的处理器。在本详细说明中,这些实现方式或本发明可采用的任何其它形式可以称为技术。通常,可以在发明的

范围内改变所公开的过程的步骤的顺序。除非另有说明,否则被描述为配置成执行任务的部件(诸如处理器或存储器)可以被实现为临时配置成在给定时间执行该任务的通用部件或者被制造为执行该任务的特定部件。如本文中所使用,术语“处理器”指代配置成处理数据(诸如计算机程序指令)的一个或多个设备、电路和/或处理核心。

[0017] 以下连同图示发明的原理的附图提供了发明的一个或多个实施例的详细描述。结合这样的实施例描述了发明,但是发明不限于任何实施例。发明的范围仅由权利要求限制,并且发明涵盖许多替换、修改以及等同物。在以下描述中阐述了许多具体细节,以便提供对发明的透彻理解。出于示例的目的而提供这些细节,并且可以在没有这些具体细节中的一些或全部的情况下根据权利要求实践发明。出于清楚的目的,没有详细描述与发明相关的技术领域中的已知的技术材料,从而使得不会不必要地模糊发明。

[0018] 本文中描述了跨多个电池特性来匹配电池(例如,在一个或多个电池组的装配期间)的技术的各种实施例。首先,描述了根据本文中描述的(一种或多种)技术而为其选择电池的电池组的示例。其次,描述了电池匹配技术的各种实施例。

[0019] 图1是图示包括多个电池的电池组的实施例的示图。在示出的示例中,电池组100包括多个电池(102),所述多个电池(102)可以根据本文描述的技术中的一种或多种进行匹配或以其它方式选择。在这个示例中,电池组包括12个电池;其它电池组实施例可以包括某个其它数量的电池。金属罐(104)用于容纳交替的电池(102)层和绝缘体(106)层。绝缘体(106)层充当阻燃剂以在电池组中的电池中的一个引燃的情况下减缓火从电池到电池的蔓延。薄片(108)被连接到电池并用于将由电池生成的功率运送到电池组之外。薄片的顶部被连接到盖子(未示出)的底部,所述盖子附接到罐的顶部。盖子的顶部具有从其可以使用由电池组中的电池生成的功率的正端和负端。

[0020] 当电池组中的所有电池跨多个所关心的电池特性匹配时,如此处所示出的包括多个电池的电池组执行地更好。在电池组装配期间,对未匹配或未分组的电池的组执行多特性电池匹配过程(所述多特性电池匹配过程的一些示例在下面被更详细地描述),使得要包括在电池组中的电池被标识。例如,假设通过电池匹配过程运行100个电池。如果该过程标识12个电池的N个组,那么将装配N个电池组并且将存在 $100 - (12N)$ 个剩余电池。电池组制造商可能不想将来自不同批次或制造系列的电池彼此混合,因此将剩余电池的数量最小化可能是期望的,这是因为任何剩余的电池可能被丢弃。

[0021] 下图描述了在随后的示例中用于匹配电池的一些示例性电池特性。

[0022] 图2是图示用于匹配电池的电池特性的实施例的表格。在本文中描述的示例中,容量(例如,以Ah为单位)、开路电压(例如,以伏特为单位)、电阻(例如,以欧姆为单位)以及自放电速率(例如,以 $\mu\text{A}$ 为单位)的电池特性被用于匹配电池。容量、开路电压、电阻以及自放电速率的值分别在行200-203中针对六个示例性电池而示出。当然,这些电池特性仅是示例性的并且本文中描述的技术可以合并电池特性的任何组合。在本文中描述的示例中,假设已经测试电池以获得所有所关心的电池特性的值,并且为简洁起见,本文中未描述测量技术。类似地,为简化和简洁起见,此表格仅示出了六个电池。当然,真实世界的电池组装配过程可以对大得多的电池的池执行电池匹配。

[0023] 在电池匹配过程期间的一个目标是要确保给定电池组中的所有电池是被匹配的。更具体地,如果每个特性之间的差在某个对应容差之内,那么两个电池被认为是匹配(有时

称为成对匹配(pair match))。例如,对于表格中的要被认为匹配的电池1(参见列204)和电池2(参见列206)而言:

[0024]  $|C1-C2|$  不应超过  $C_{容差}$ ; 并且

[0025]  $|V1-V2|$  不应超过  $V_{容差}$ ; 并且

[0026]  $|R1-R2|$  不应超过  $R_{容差}$ ; 并且

[0027]  $|D1-D2|$  不应超过  $D_{容差}$ 。

[0028] 如果即使电池特性中的一个具有超过对应容差的差,那么那个电池对(pair of cell)不是匹配(即,它们不是兼容的)且不应将它们放于相同电池组中。当然,该对不兼容电池可被分开并且(如果可能的话)被包括在不同电池组中。

[0029] 遗憾的是,特性不是强相关的并且电池对可能根据一个特性是兼容的而根据另一特性不是兼容的。下图示出了这方面的示例。

[0030] 图3是图示电池特性的分布的实施例的示图。在示出的示例中,示图300-303分别示出了容量、开路电压、电阻以及自放电速率的示例分布。如所示出的,所有分布具有不同形状。

[0031] 示图还示出了两个示例性电池落入分布内的什么地方。利用星形表示电池A而利用三角形表示电池b。关于容量(参见示图300)和自放电速率(参见示图303),因为差值未超过相关容差,所以两个电池是兼容的。换言之,关于那些特性它们彼此是足够靠近的。然而,关于开路电压(参见示图301)和电阻(参见示图302),因为那些特性的差值超过相关容差,所以电池是不兼容的。换言之,对于那些特性值离得太远。

[0032] 下图描述了在一些实施例中使用图2中所示出的四个示例性电池特性的示例性电池匹配过程。

[0033] 图4是图示用来跨多个电池特性执行电池匹配的过程的实施例的流程图。在一些实施例中,该过程在电池组装配过程期间执行,在电池组装配过程中来自相同电池制造系列或批次的成百上千的电池跨多个所关心的电池特性而被评估并且被标识以用于包含在电池组中。

[0034] 在400,针对多个电池中的每个电池,接收第一电池特性以及第二电池特性以便获得多个电池特性。当然,可以使用任何数量的所关心的特性来执行电池匹配过程,并且可以获得额外的电池特性。在图2的示例中,对于六个电池接收四个电池特性,并且那里示出的表格是可在步骤400接收的多个电池特性的一个示例。

[0035] 在402,对于多个电池中的每个电池,至少部分地基于多个电池特性确定批次兼容性数量,其相关联于与那个电池兼容的兼容电池的数量。在一些实施例中,步骤402包括生成电池兼容性矩阵,所述电池兼容性矩阵指示(例如,针对所有可能的电池对)给定电池对是否是跨第一电池特性、第二电池特性以及任何其它所关心的电池特性(如果适用的话)兼容的(并且因此可被一起放于相同电池组中)。在一些实施例中,批次兼容性数量在计算批次兼容性数量时计算电池与其自身的兼容性。在一些其它实施例中,未对那个电池的批次兼容性数量计算电池与其自身的兼容性。下图示出了示例电池兼容性矩阵。

[0036] 图5是图示电池兼容性矩阵的实施例的示图。在所示出的示例中,兼容性矩阵500中每个条目指示特定的电池对是否是彼此兼容的。沿对角线的条目对应于给定电池与其自身的比较,且因此对于那些条目示出了“不适用”。当然,在一些实施例中,过程可以选择利



用兼容指示或不兼容指示来填充沿电池兼容性矩阵的对角线的条目(例如,在下文的伪代码示例中,填充电池兼容性矩阵使得电池*i*与电池*i*(即,其自身)是兼容的)。由于(电池*i*, 电池*j*)的条目必须与(电池*j*, 电池*i*)的条目相同,因此表格沿对角线是对称的,并且此性质可用于填充*n*×*n*表格中的条目中的一些(例如,而不用再次执行相同的兼容性检查)。在一些实施例中,*n*×*n*表格(诸如所示出的表格)被压缩成更小的东西,这是因为该*n*×*n*表格包含重复信息。

[0037] 检查所有关心的电池特性的成对电池兼容性测试可用于填充所示出的示例性电池兼容性矩阵。下图描述了其中存在两个关心的电池特性的一个示例。

[0038] 图6是图示用于成对电池兼容性测试的过程的实施例的流程图。在一些实施例中,示例过程被用于确定图4中的步骤402的兼容电池的数量。在一些实施例中,示例过程被用于填充电池兼容性矩阵,诸如图5中所示出的电池兼容性矩阵。在所示出的示例中,基于两个关心的电池特性来评估兼容性。当然,可以在所示出的示例性过程之上进行扩展(例如,通过重复某些步骤),以便跨三个或更多关心的电池特性来评估电池对。

[0039] 在步骤600,针对第一电池特性确定多个电池中的第一电池与多个电池中的第二电池之间的第一差。在步骤602,针对第二电池特性确定第一电池与第二电池之间的第二差。例如,在上面的等式中, $|C1-C2|$ (备选地, $C1-C2$ )是步骤600的示例,且 $|V1-V2|$ (备选地, $V1-V2$ )是步骤602的示例。

[0040] 在步骤604,确定第一差是否超过与第一电池特性相关联的第一容差。例如, $C_{容差}$ (来自上面的等式)是第一容差的示例。如果在步骤604确定第一差超过第一容差,那么在步骤608宣布第一电池与第二电池是不兼容的。

[0041] 如果在步骤604确定第一差不超过第一容差,那么在步骤606确定第二差是否超过与第二电池特性相关联的第二容差。例如, $C_{容差}$ 是来自上面的等式的第二容差的示例。如果在步骤606确定第二差超过第二容差,那么在步骤608宣布第一电池与第二电池是不兼容的。否则,在步骤610宣布第一电池与第二电池是兼容的。换言之,为了两个电池是兼容的,所有关心的电池特性的差值必需在其相应容差内。

[0042] 以下示出了其中以伪代码生成(例如,未整理的)电池兼容性矩阵的另一示例:

```
Define nCells as the number of cells in the batch

Define a [nCells x nCells] matrix M, the unsorted
compatibility matrix

/*贯穿每个电池而循环，针对与批次中的所有其它电池的兼容性进行检查 */

For each cell k:
  For each cell j:
    If cell k's attributes are within tolerances of
cell j's attributes:
      M(k,j) = 1
    else
      M(k,j) = 0
    end
  end
end

end
```

[0043]

/\*这将生成具有每个电池与每个其它电池的兼容性的矩阵。如果电池 k 与电池 j 是兼容的，那么 M (k, j) 为 1，并且否则 M (k, j) 为 0\*/

[0044] 回到图4，一旦在步骤402已经确定批次兼容性数量，则在步骤404根据批次兼容性数量整理多个电池以便获得电池的整理的列表。下图示出了电池的整理的列表的示例。

[0045] 图7是图示电池的整理的列表的实施例的示图。在这个示例中，使用来自图5的电池兼容性矩阵生成整理的列表 (700)。在这个示例中按降序整理列表，使得电池6 (其与四个其它电池是兼容的) 列于第一，电池4 (其与四个其它电池是兼容的) 列于第二，电池2 (其与三个其它电池是兼容的) 列于第三，电池1 (其与三个其它电池是兼容的) 列于第四，电池5 (其与两个其它电池是兼容的) 列于第五，并且电池3 (其与两个其它电池是兼容的) 列于第六。这是在图4中的步骤404生成的整理的列表的一个示例。尽管图7的示例从底部向顶部 (即，按升序) 地评估列表，但本文中所描述的技术也利用按升序整理的列表而起作用。

[0046] 下面示出了其中以伪代码生成整理的电池兼容性矩阵 (其是电池的整理的列表的一个示例) 的另一示例：

/\*通过将 M 的它的对应行相加来计算每个电池 k 的 batchCompatibilityScore。\*/

```
Define a [nCells x 1] vector batchCompatibilityScore
```

```
for each cell k = 1:nCells:
```

```
    batchCompatibilityScore(k) = sum(M(k,:))
```

```
end
```

```
Define a function [Y,I] = sort(X) such that:
```

```
    X is a vector of values
```

```
    Y is the elements of X, sorted in ascending order
```

```
    I is a vector of indices such that X(I) = Y
```

```
Define a [nCells x 1] vector Cmap as follows:
```

[0047]

```
    [~,Cmap] = sort(batchCompatibilityScore)
```

/\*因此，元素 Cmap(1)是具有最低的 batchCompatibilityScore 值的电池的索引\*/

```
Define a [nCells x nCells] matrix C, the sorted compatibility matrix
```

/\*贯穿 C 的每行逐步操作。使用 Cmap 作为映射关键，从 M 矩阵到 C 矩阵通过对应值进行映射。\*/

```
For each cell k = 1:nCells:
```

```
    C(k,:) = M(Cmap(k),Cmap)
```

```
end
```

/\*排列所得到的 C 矩阵，使得随着 C 的行/列数增加，与那些行对应的电池将具有更高的批次兼容性。由 C 矩阵中的行 k 表示的电池对应于矩阵 M 中的第 Cmap(k) 行。\*/

[0048] 回到图4,在406,生成要包括在电池组中的兼容电池的列表,其包括通过根据由电池的整理的列表所指定的顺序来评估多个电池并且以具有最低批次兼容性数量的电池开始。这方面的示例被示出在下图中。

[0049] 图8是图示根据由电池的整理的列表所指定的顺序来评估的电池的实施例的示意图。这个示例继续先前的图的示例。最初,要包括在电池组中的兼容电池的集合是空集(例如, {})。具有最低批次兼容性数量的电池首先被评估,并且被(例如,自动地)包括在要包括在电池组中的兼容电池的集合中。在这个示例中,这意味着电池3被包括在要包括在电池组中的电池的集合中(例如,参见图7,其中电池3处于整理的列表的底部)。

[0050] 一旦要包括在电池组中的兼容电池的集合非空,则针对集合中的所有电池评估任何正被评估的电池。示图800示出这方面的第一次迭代。左列(802a)示出兼容电池的当前集合,其在所示出的状态中仅包括电池3。右列(804a)示出了整理的列表,其被更新以反映已被选择要被包括在电池组中的任何电池。同样地,更新的整理的列表804a不包括电池3,这是因为其已被选择以用于包括在电池组中。

[0051] 在示图800中,电池5(即,正被评估的电池)针对电池3(即,要包括在电池组中的兼容电池的当前集合)被比较。在一些实施例中,查询电池兼容性矩阵以查看两个电池是否彼此兼容。如图5中所示出的,电池5与电池3彼此不兼容,并且因此不将电池5从右列移至左列。

[0052] 示图806示出了正被评估的更新的整理的列表中的下一个电池。在此迭代中,检查

电池3(参见左列802b)与电池1(参见右列804b)的兼容性。按照图5中所示出的电池兼容性矩阵,那两个电池是兼容的,因此将电池1从右列移至左列。

[0053] 示图808示出了下一次迭代。在此状态下,电池2(参见右列804c)正被评估并且相对电池3以及电池1(参见左列802c)被比较。电池2与两个电池都不兼容,因此不将电池2从右列移至左列。为了要被包括在左边的集合中,右列中正被评估的电池需要与左列中的所有电池是兼容的。

[0054] 一旦已经标识足够的兼容电池来填装电池组,过程(至少暂时地)结束并且所标识的电池被包括在被装配的下一个电池组中。在一些实施例中,一旦被选择,兼容电池在电池组内不必遵循任何特定的顺序,并且可以使用任何适当的技术来将电池组中的电池封装或分层。然后可以在整理的列表中的剩下的电池上重复电池匹配过程以标识要包括在下一个电池组中的兼容电池。

[0055] 在伪代码中,图4中的步骤406的另一示例是:

```
Define a [nCells x nCells] matrix Cremnant such that:
```

```
Cremnant = C;
```

```
/*以整理的顺序 Cremnant 表示尚未被匹配到任何集合的剩下的电池。*/
```

```
Define a function [validSet] = findValidSet(iCell,X) such that:
```

```
X is a compatibility matrix where each element (k,j) is cell k's compatibility with cell j. Compatibility is expressed as a 1 if compatible and a 0 if not.
```

```
iCell is a cell index.
```

```
validSet is a [12 x 1] vector where validSet(1) = iCell and validSet(2:12) are the 11 lowest indices in matrix X corresponding cells that are compatible with both the cell denoted by iCell and each other. The function returns validSet = NULL if there are no valid sets possible.
```

[0056]

```
For each cell k = 1:nCells
```

```
/*检查此电池是否与至少 11 个其它电池是兼容的*/
```

```
If sum(Cremnant(k,:) >= 11)
    validSet = findValidSet(k,Cremnant)
    If a valid matching set was found:
        iSet = iSet + 1
        sets(iSet,:) = validSet
```

/\*从 Cremnant 矩阵移除对应于刚刚形成的集合中的所有电池,因此它们不会被再使用。\*/

```
Cremnant(validSet,:) = 0
```

```
Cremnant(:,validSet) = 0
```

```
end
```

```
end
```

```
end
```

[0057] 如果期望的话,下文包含有关validSet=findValidSet(iCell,X)函数的更多信息:

[0058]

```
As previously described in the sorting methodology:
```

X is a compatibility matrix where each element (k,j) is cell k's compatibility with cell j. Compatibility is expressed as a 1 if compatible and a 0 if not.

iCell is a cell index.

validSet is a [12 x 1] vector where validSet(1) = iCell and validSet(2:12) are the 11 lowest indices in matrix X corresponding cells that are compatible with both the cell denoted by iCell and each other. The function returns validSet = NULL if there are no valid sets possible.

The function will choose the cells with the lowest indices that can form a matching set. Thus, this function relies on the fact that X is a sorted compatibility matrix in order to form a set with the least batch-compatible cells. The implementation is as follows:

/\*初始化\*/

Define a vector matchList as the list of indices of all cells that are compatible with cell iCell, sorted in ascending order.

Define a [1 x 12] vector tempSet, container of cell indices for prospective sets.

tempSet(1) = iCell  
tempSet(2:12) = 0

[0059]

/\*此计数器记录我们预期集合的大小\*/

iSet = 2

/\*此计数器在添加新预期集合成员时跟踪我们在 matchingCells 列表中的什么地方。\*/

iSearch = 1

/\*此计数器跟踪其中我们开始搜索的 matchingCells 索引。\*/

iSearchStart = 1

Define a function [flag] = isValidSet(set,X):

set is a [1 x 12] vector of cell indices.

X is the compatibility matrix

flag is a boolean that is TRUE if the cells

corresponding to all non-zero indices in set are

compatible with each other and FALSE if otherwise.

/\*开始搜索有效集合\*/

/\*整个方法是查遍 (search through) 兼容电池的列表, 将它们一次一个地添加到 tempSet 容器 (container), 并检查 tempSet 中的所有集合是否仍然是有效的。如果电池的

列表被穷尽，则从 tempSet 移除除了 iCell 外的所有电池，并且从兼容电池的列表中的另一位置处重新开始该过程。如果全部开始位置已被穷尽，则在 X 中没有包含 iCell 的有效集合。\*/

```

While true:
    tempSet(iSet) = matchingCells(iSearch)

    If isValidSet(tempSet,X)

/*如果到目前为止我们具有有效的集合，则继续搜索到 tempSet 的下一个有效添加*/

        tempSet(iSet) = matchingCells(iSearch);

/*如果我们刚刚将第 12 个电池添加到集合，则我们已经完成*/

        If iSet == 12
            validSet = tempSet;
            return
        End

        iSet = iSet + 1

```

[0060]

```

end

/*我们尚未完成，那么增大 iSearch 以进一步向下搜查 matchingCells 列表*/

iSearch = iSearch + 1;

/*其它退出条件*/

If iSearch > number of elements of matchingCells

/*我们已经用完用来查遍的电池并且还没有找到有效集合。但在 matchingCells 上的一个更远的索引处再开始搜索。*/

    iSet = 1;
    tempSet(2:12) = 0
    iSearchStart = iSearchStart + 1 /*增大我们的搜索起点*/

    iSearch = iSearchStart /*重置我们在 matchingCells 中的搜索位置*/

end

If iSearchStart > number of elements in matchingCells

/*如果 iSearchStart 超过 matchingCells 的长度，我们已经开始在每个位置的搜索，并且没有找到任何有效的集合。我们可以取消搜索。*/

    validSet = NULL
    return

```

[0061]

```

end
end

```

[0062] 回到图7，如那里所示出的，在一些情况下将存在具有相同批次兼容性数量的电池（即，平局）。例如，电池6与电池4彼此成平局、电池2与电池1彼此成平局，以及电池5与电池3彼此成平局。下图描述了在两个或更多电池具有相同批次兼容性数量时的各种决胜局示例。

[0063] 图9是图示决胜局过程的实施例的流程图。在一些实施例中,当生成电池的整理的列表时,在图4中的步骤404对平局电池的任何组执行过程。本文中所描述的此决胜局过程以及其它决胜局过程可以如所需要的针对平局电池的每个组进行重复(例如,在不考虑其它组中的其它平局电池和/或非平局的电池的情况下)。

[0064] 在900,针对多个平局电池中的每个平局电池确定兼容电池的集合,以便获得兼容电池的多个集合。例如,在图7中,电池6与电池4彼此成平局,这是因为其二者都具有为4的批次兼容性数量。应用到那些平局电池的步骤900将确定哪些电池与电池6是兼容的以及哪些电池与电池4是兼容的。

[0065] 在902,至少部分地基于与兼容电池的多个集合相关联的唯一性以及兼容电池的多个集合相关联的一个或多个批次兼容性数量来打破多个平局电池之间的平局。为了继续来自上面以及图7的示例,将在打破电池6与电池4之间的平局中考虑与下列两项相关联的唯一性以及批次兼容性数量:与电池6兼容的电池的集合以及与电池4兼容的电池的集合。

[0066] 在一些实施例中,在步骤902全局地考虑或评估唯一性。例如,可以对兼容电池的多个集合执行全局唯一化过程,使得任何相同项(duplicate)被移除。备选地,可以局部地(例如在兼容电池的整理的列表或排序的列表内的索引或者的特定级别的上下文中)评估唯一性。下图示出了两者的一些示例。

[0067] 图10是图示决胜局技术的实施例的示图,所述决胜局技术首先全局地使与平局电池兼容的电池唯一化,并且然后检验唯一化的兼容电池的批次兼容性数量。这是可如何执行图9中的步骤902的一个示例。

[0068] 示图1000示出了在决胜局过程的开始时的三个示例性平局电池。在此示例中,所有平局电池具有为4的批次兼容性数量,这是因为它们与四个其它电池兼容。第一平局电池(参见行1002a)与电池1、电池5、电池7以及电池8是兼容的。第二平局电池(参见行1004a)与电池5、电池7、电池8以及电池10是兼容的。第三平局电池(参见行1006a)与电池1、电池4、电池8以及电池9是兼容的。

[0069] 通过移除任何相同项而将与平局电池兼容的电池(在示图1000中的右列中示出)全局地唯一化(例如,跨所有集合/行)。由于电池1、电池5、电池7以及电池8在示图1000中出现不止一次,因而那些电池被移除并且示图1010示出了没有那些重复的电池的集合/行。行1002b(与第一平局电池相关联)现在是空集,行1004b(与第二平局电池相关联)现在包括电池10,并且行1006b(与第三平局电池相关联)现在包括电池4和电池9。

[0070] 然后检验唯一化的兼容电池的批次兼容性数量。示图1010以及示图1012在其右列中示出了相同的唯一化的兼容电池,但是示图1012还在括号中示出了对应批次兼容性数量。基于针对每一行的最低批次兼容性数量(如果有的话)来打破平局。行1006c(与第三平局电池相关联)具有两个电池/分值,但是那行中的最低批次兼容性数量是为9的批次兼容性数量(对应于电池4)。由于为9的那个批次兼容性数量(对应于电池4)不比行1004c中为7的批次兼容性数量(对应于第二平局电池)更低,因而首先选择第二平局电池。就是说,这个示例中的第二平局电池将首先在整理的列表中被列出,并且在此组中的其它平局电池之前被评估以用于包括在电池组中。其次(即,第二)将选择第三平局电池,这是因为它具有次低批次兼容性数量(即,针对那行的为9的批次兼容性数量)。

[0071] 示图1010/1012中的行1002b/1002c为空,并且因此不存在用来示出或检验的对应批次兼容性数量。在移除了相同项之后的这样的空集可以在决胜局过程期间最后被列出或选择(例如,这是因为它们兼容于其又与组中的其它平局电池兼容的电池,并且因此相比于组中的其它平局电池不存在那些平局电池可以做出的唯一或特定匹配)。同样地,决胜局过程将(最终)第一具有第二平局电池、第二具有第三平局电池,以及第三具有第一平局电池。

[0072] 下图在流程图中更正式地和/或更一般地描述了这方面。

[0073] 图11是图示决胜局过程的实施例的流程图,所述决胜局过程首先全局地使与平局电池兼容的电池唯一化,并且然后检验唯一化的兼容电池的批次兼容性数量。在一些实施例中,图9中的步骤902包括所示出的过程。

[0074] 在1100,使与多个平局电池相关联的兼容电池的多个集合全局地唯一化,以便获得兼容电池的多个唯一化的集合。例如,参见图10中的示图1000与示图1010之间的相同项的移除。

[0075] 在1102,针对多个唯一化的集合中的每个非空唯一化的集合确定最低批次兼容性数量。例如,在图10中的示图1012中,这个步骤不适用于行1002c,这是因为那行为空。针对示图1012中的行1004c,最低批次兼容性数量是7,并且针对行1006c,最低批次兼容性数量是9。

[0076] 在步骤1104,至少部分地基于每个非空唯一化的集合的最低批次兼容性数量来整理多个平局电池,其中当打破平局时任何空的唯一化的集合最后被排列。例如,参见图10中的示图1012中示出的排列。注意通过决胜局过程首先排列行1004c(因为它具有为7的最低批次兼容性数量),并且其次排列行1006c(因为它具有为9的次低批次兼容性数量)。通过决胜(tiebreaking)过程最后排列行1002c(其在唯一化之后成为空集)。

[0077] 要注意的是图10和图11中所描述的技术不一定是最佳仲裁(tie breaking)技术。例如,在图10中,如果电池1具有最低批次兼容性数量,则想到应当将第一平局电池或第三平局电池(而不是第二平局电池)排行第一。下面描述更好的仲裁技术(例如,根据性能),但如下面将更加详细地描述的那样,以下技术的步骤或迭代的数量是可变的(其可能使得运行时间更长,而上面描述的技术具有固定数量的步骤),并且复杂性的程度更高(例如,在下面的伪代码中被递归调用的findTies函数)。对于一些应用,性能提高不值得潜在更长的运行时间和/或增加的复杂性,并且使用上面描述的技术。在其它应用(例如,其中性能提高是值得上面描述的代价的)中,可以使用下面描述的技术。在下面的示例中,在兼容电池的整理的列表或排列的列表中的特定级别的上下文中评估在图9中的步骤902的唯一性。以下伪代码示出了这方面的示例。



```
Define a function [C_noties, Cmap_noties] =  
tiebreaker(C,Cmap) such that  
C is the compatibility matrix
```

```
Cmap is a vector of indices of the original set of  
cells, sorted by compatibility score  
C_noties is the compatibility matrix with a secondary  
sorting process applied in order to break up ties  
between cells with equal batchCompatibilityScore values  
Cmap_noties is the revised mapping matrix that maps a  
the kth cell of C_noties with the index of the original  
set of cells
```

[0078]

```
Define nCells as the number of cells in the batch (the size  
of one dimension of C)
```

```
/*找到 C 的 batchCompatibilityScore 值*/
```

```
for each cell k = 1 :nCells:  
    batchCompatibilityScore_sorted(k) = sum(C(k,:))  
end
```

```
/*注意，由于 C 是被整理的，因而 batchCompatibilityScore_sorted 已经按上升的值  
被整理*/
```

```

/*暂时分配输出变量*/

Cmap_noties = Cmap
C_noties = C

/*找到所有平局*/

Define a function [Y] = findTies(X) such that:
  X is a vector of all batchCompatibilityScore values
  Y is a set of arrays such that each array contains the
  indices of each cell that has the same
  batchCompatibilityScore with other cells in that array.

Example (note that all indices are 1-indexed): if X = [1
2 3 3 3 4 5 6 6 7] then Y = findTies(X) returns:
Y = [[3 4 5], [8, 9]]
And using cell array notation, Y{1} = [3, 4, 5] and Y{2}
= [8, 9]

ties = findTies(batchCompatibilityScore_sorted)

Define numTies as the number of arrays in ties

[0079] for each tie k = 1:numTies
        tiedCells = ties{k}

        /*找到 tiedCells 中的每个电池中的最不兼容的电池*/

        minCell = min(C(tiedCells,:))

        /*如果存在具有相同 minCell 值的多个电池，那么存在另一平局，findTies
function 必须递归地在剩下的平局电池上运行直到每个值 minCell 是 C(tiedCells,:)中的
最低索引的电池（在平局的适当级别下对于其与之成平局的所有其它电池是唯一的）为止。
*/

        /*将 C 更新为按照新顺序*/

        C_noties(ties{k},:) = C(ties{k}(iTiebreakOrder),:)

        Define numTiedCells as the number of elements in
        tiedCells
        for each cell j in 1:numTiedCells:
            Cmap_noties(ties{k}(j)) =
            Cmap_noties(ties{k}(iTiebreakOrder(j))
        end
end

```

[0080] 下图示出了使用示例性批次兼容性数量的上面的伪代码的示例。

[0081] 图12是图示决胜局技术的实施例的示图，其中与平局电池兼容的电池基于批次兼容性数量被排序，并且然后在逐级别的基础上被评估。

[0082] 示图1200示出了在决胜局过程的开始时的三个示例性平局电池。所有三个平局电池具有为4的批次兼容性数量，并且因此成平局。第一平局电池（参见行1202）与电池A、电池B、电池C以及电池D兼容；第二平局电池（参见行1204）与电池B、电池C、电池E以及电池F兼容；第三平局电池（参见行1206）与电池B、电池C、电池D以及电池E兼容。

[0083] 所有兼容电池（示出在右列中）具有在括号中示出的它们的批次兼容性数量。兼容电池的每个集合（例如，行1202、行1204以及行1206）基于批次兼容性数量被整理以产生兼

容电池的整理的列表。注意,例如按从左进行到右的上升的批次兼容性数量整理行1202、行1204以及行1206中的兼容电池。

[0084] 在基于批次兼容性数量整理兼容电池的集合的情况下,从左进行到右(即,按上升的批次兼容性数量顺序)在逐级别的基础上执行检查。检查确定最小批次兼容性数量是否是唯一的(例如,针对那个级别以及针对正被评估的批次兼容性数量或电池)。示图1200示出了在第一(即,最低的)级别下的第一检查。在那个级别下,最低批次兼容性数量是1(与第一平局电池的电池A相关联),并且那个批次兼容性数量是唯一的,因此第一平局电池赢得那个决胜局回合。

[0085] 示图1208示出了在第二低的级别下的第二检查。由于第一平局电池已经由决胜局过程所选择或挑选,因而第一平局电池不是此处检查的部分。注意,例如它的第一名的排名被示出到示图1208中的行1202的左侧。在此检查中,最小批次兼容性数量是3,但是那个批次兼容性数量不是唯一的,因此决胜过程继续前进到下一个级别而不宣布获胜者。

[0086] 示图1212示出了在第三低的级别下的第三检查。最低批次兼容性数量是4(与第三平局电池的电池D相关联)并且那个批次兼容性数量是唯一的。因此,第三平局电池赢得此决胜局回合。通过排除,第二平局电池是第三/最后。由示例性决胜局过程所产生的最终排名被示出到示图1210中的表格的左侧:第一平局电池,然后第三平局电池,并且最后第二平局电池。如此处所示出的,唯一性可以被局部地考虑在电池的整理的列表或排列的列表中的级别的上下文中(例如,在不考虑其它级别的情况下)和/或在被比较的电池的上下文中(例如,在不考虑不是比较的部分的其它电池的情况下)。

[0087] 下图在流程图中更正式地和/或更一般地描述了这方面。

[0088] 图13是图示决胜局过程的实施例的流程图,其中与平局电池兼容的电池基于批次兼容性数量被整理,并且然后在逐级别的基础上针对唯一性被评估。在一些实施例中,图9中的步骤900包括所示出的过程。

[0089] 在1300,至少部分地基于与兼容电池的多个集合相关联的批次兼容性数量来整理兼容电池的多个集合以获得兼容电池的多个整理的列表。例如,参见图12中的示图1200。行1202、行1204以及行1206按上升的批次兼容性数量顺序(其中用于整理的批次兼容性数量相关联于与平局电池兼容的电池)被整理。

[0090] 在1302,从兼容电池的不同整理的列表中选择用来比较的两个或更多兼容电池,其中所选择的兼容电池在兼容电池的其相应整理的列表中的相同级别下。例如,在图12中的示图1200中,兼容电池A(来自行1202)、兼容电池B(来自行1204)、兼容电池B(来自行1206)全部在相同列中且被选择以用于比较。

[0091] 在1304,确定与用来比较的所选择的兼容电池相关联的最小批次兼容性数量是否是唯一的。例如,在图12中的示图1200中,最小批次兼容性数量是一并且它是唯一的。如果如此,与最小且唯一的批次兼容性数量相关联的平局电池被选择为决胜局获胜者。例如,在图12中的示图1200中,第一平局电池被选择为第一回合的获胜者。

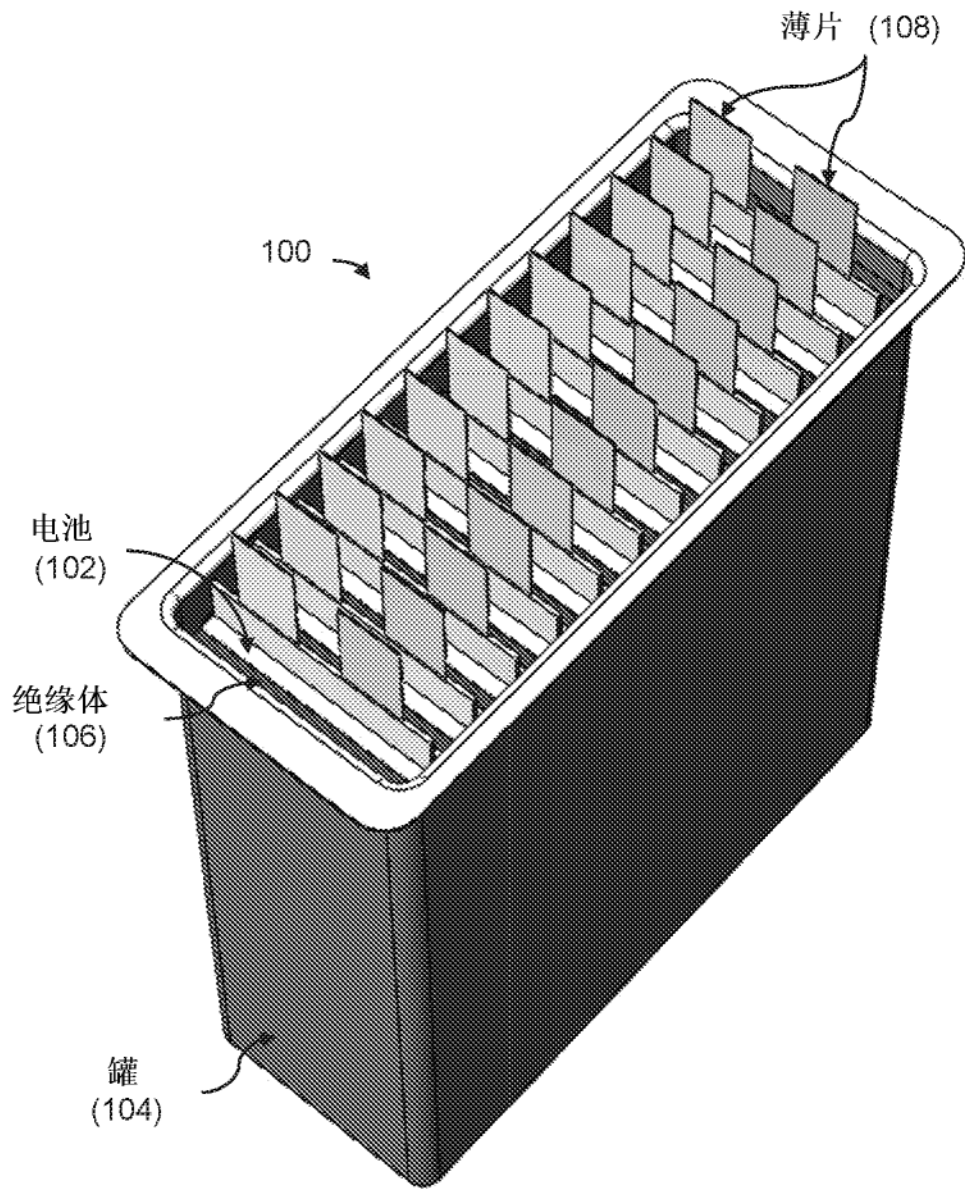
[0092] 在1308,确定是否存在更多的平局电池。在图12的示例中,第二平局电池和第三平局电池仍然具有相对于彼此的未解决的优先级或顺序,并且因此针对那种情况的在步骤1308的判定为是。如果不存在更多的平局电池,则过程结束。

[0093] 如果在步骤1308存在更多的平局电池,在步骤1302选择用来比较的两个或更多的

兼容电池。例如，在图12中的示图1208中，已经赢得决胜回合的任何平局电池不在被考虑（例如，第一平局电池不再被考虑）并且所选择的兼容电池与下一个级别相关联（例如，比较从最低的级别移至第二低的级别）。

[0094] 如果在步骤1304确定与用来比较的所选择的兼容电池相关联的最小批次兼容性数量不是唯一的，则在步骤1302选择用来比较的两个或更多兼容电池。例如，参见图12中的示图1208。在那个比较中，最小批次兼容性数量为三并且它不是唯一的。因此，针对那个回合不宣布获胜者并且比较继续前进到下一个级别（在那个示例中，第二高的级别）以及对应列中的兼容电池。

[0095] 虽然为了清楚理解的目的在一些细节中已经描述了前述的实施例，但是发明不限于所提供的细节。存在实现发明的许多备选方式。所公开的实施例是说明性的而非限制性的。



电池组 (没有盖子)

图 1

	204 电池1	206 电池2	电池3	电池4	电池5	电池6
200 容量	C1	C2	C3	C4	C5	C6
201 开路电压	V1	V2	V3	V4	V5	V6
202 电阻	R1	R2	R3	R4	R5	R6
203 自放电速率	D1	D2	D3	D4	D5	D6

电池特性值

图 2

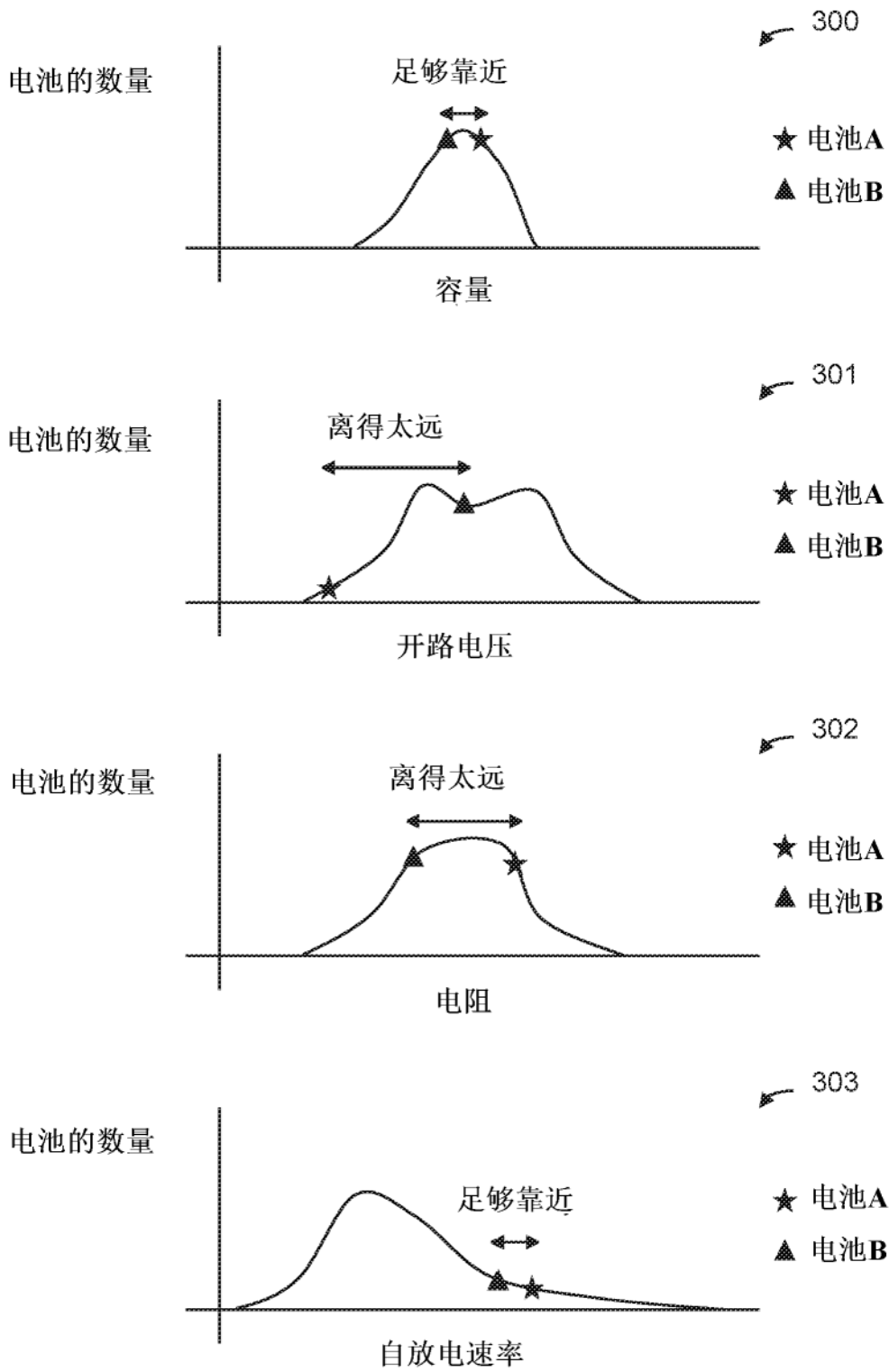


图 3

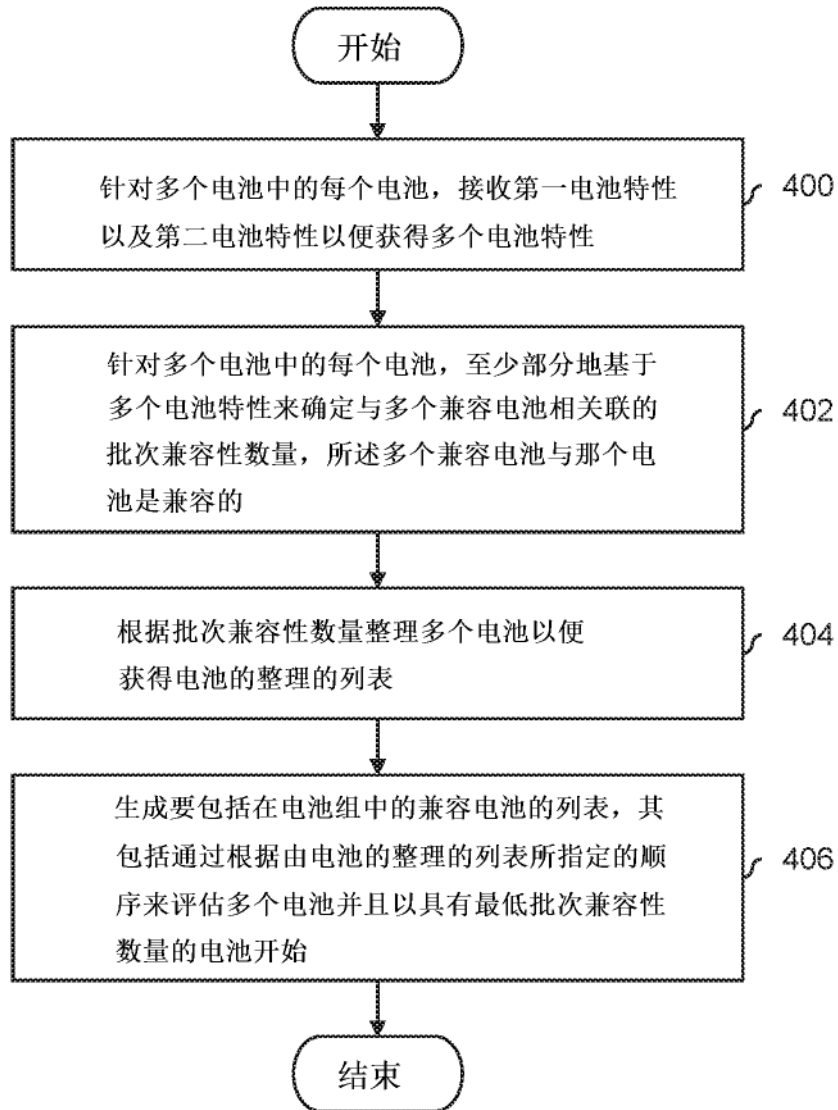


图 4



500

	电池1	电池2	电池3	电池4	电池5	电池6
电池1	N/A	IC	C	C	IC	C
电池2	IC	N/A	IC	C	C	C
电池3	C	IC	N/A	C	IC	IC
电池4	C	C	C	N/A	IC	C
电池5	IC	C	IC	IC	N/A	C
电池6	C	C	IC	C	C	N/A

电池兼容性矩阵  
(C = 兼容; IC = 不兼容)

图 5

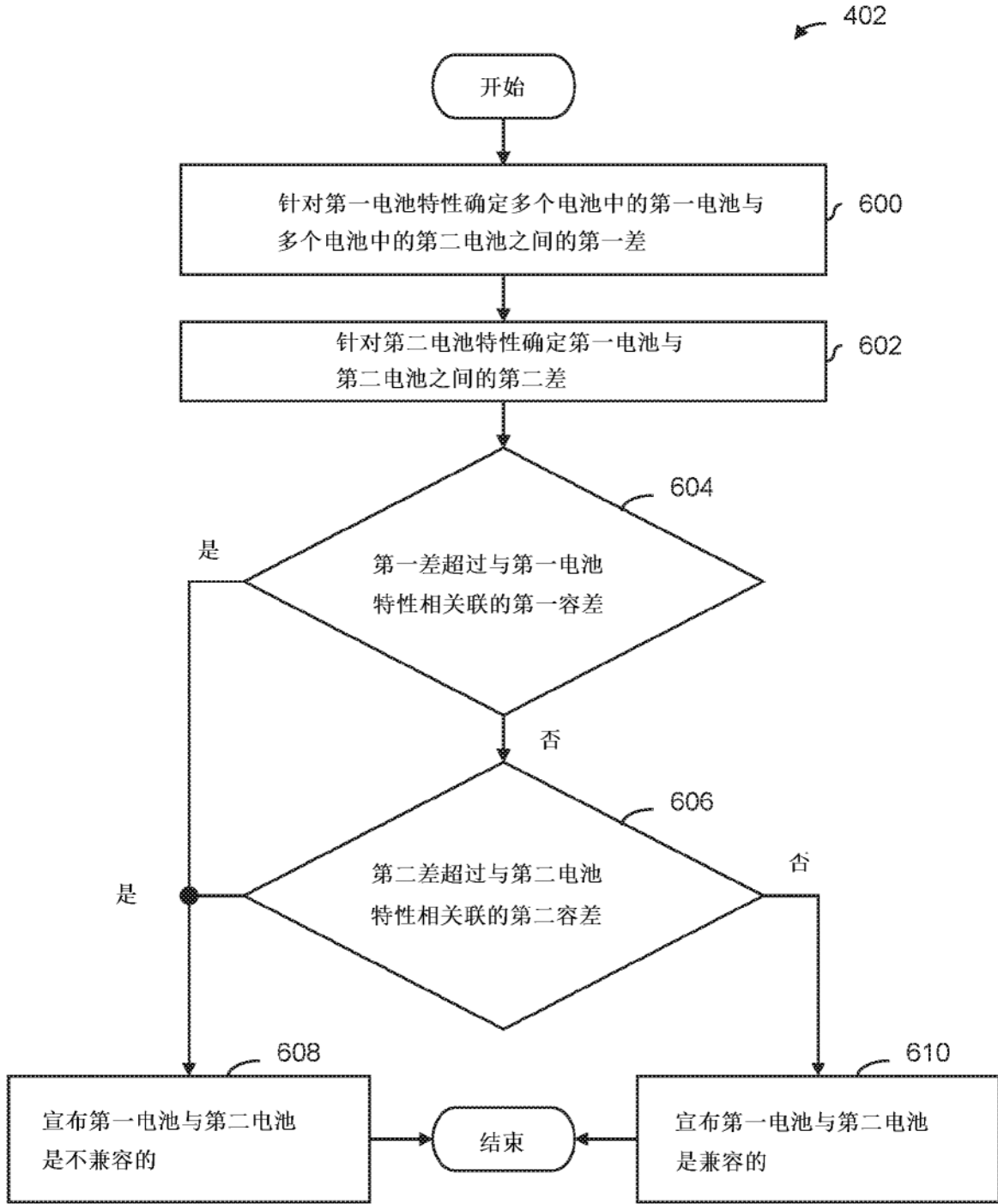


图 6

700

	批次兼容性数量
电池6	4
电池4	4
电池2	3
电池1	3
电池5	2
电池3	2



从底部向顶部地评估

电池的整理的列表

图 7

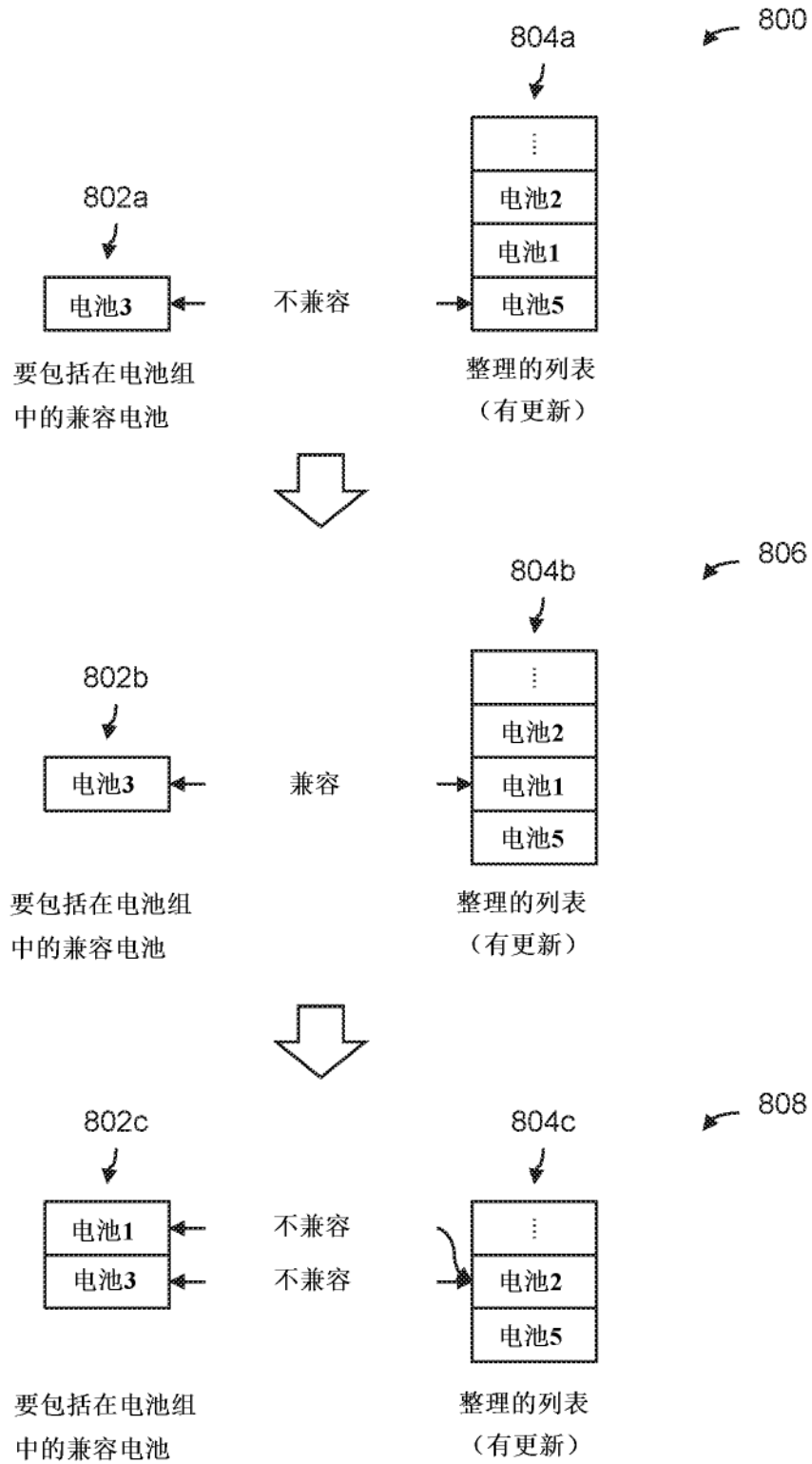


图 8

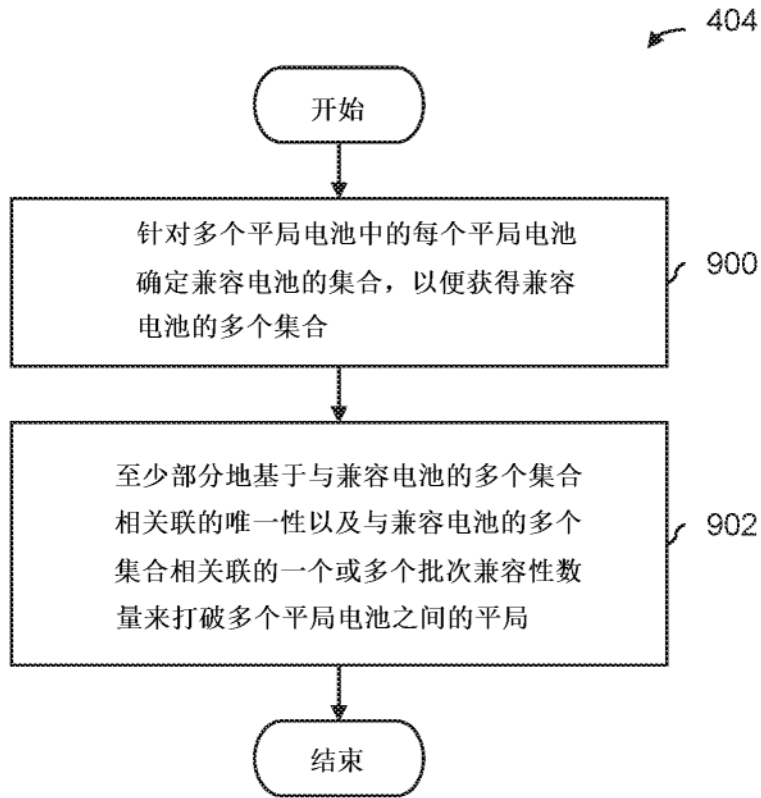


图 9

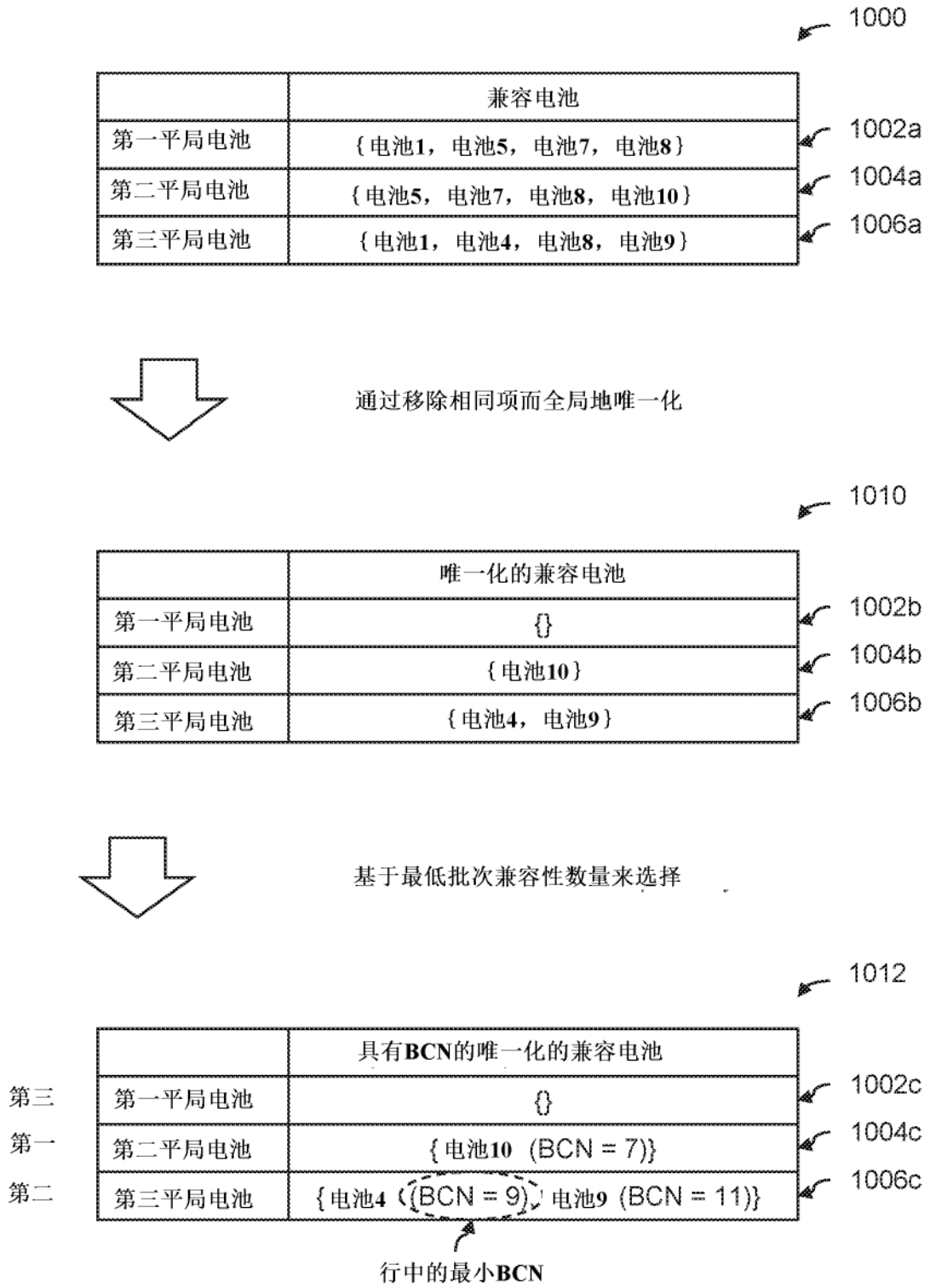


图 10

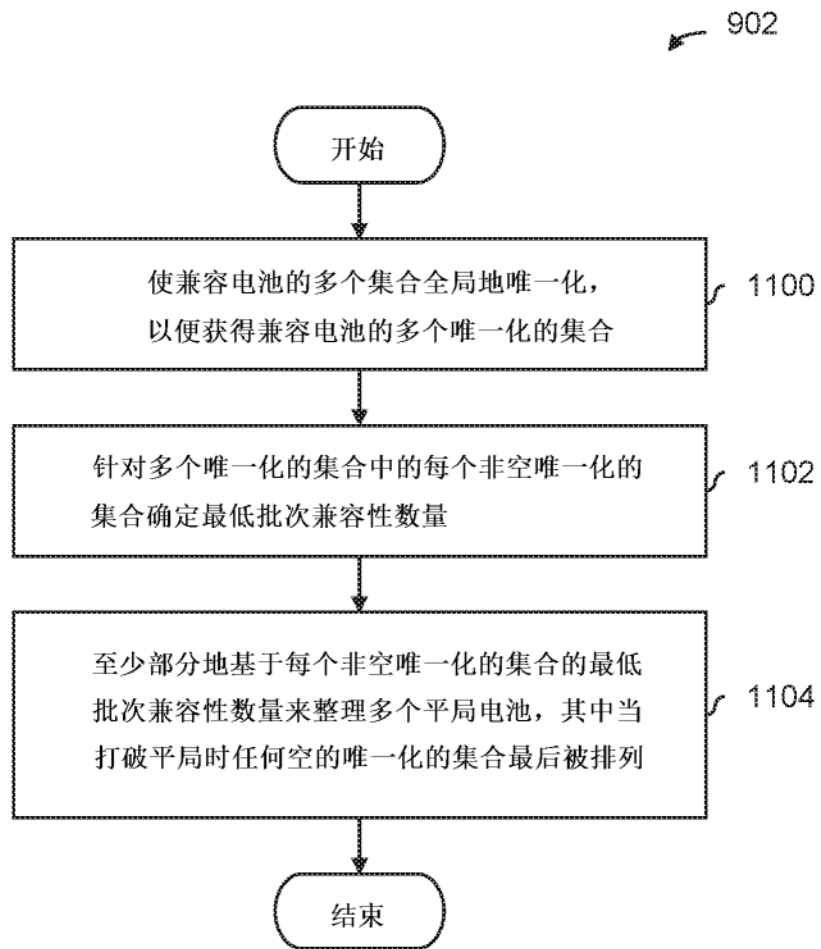


图 11

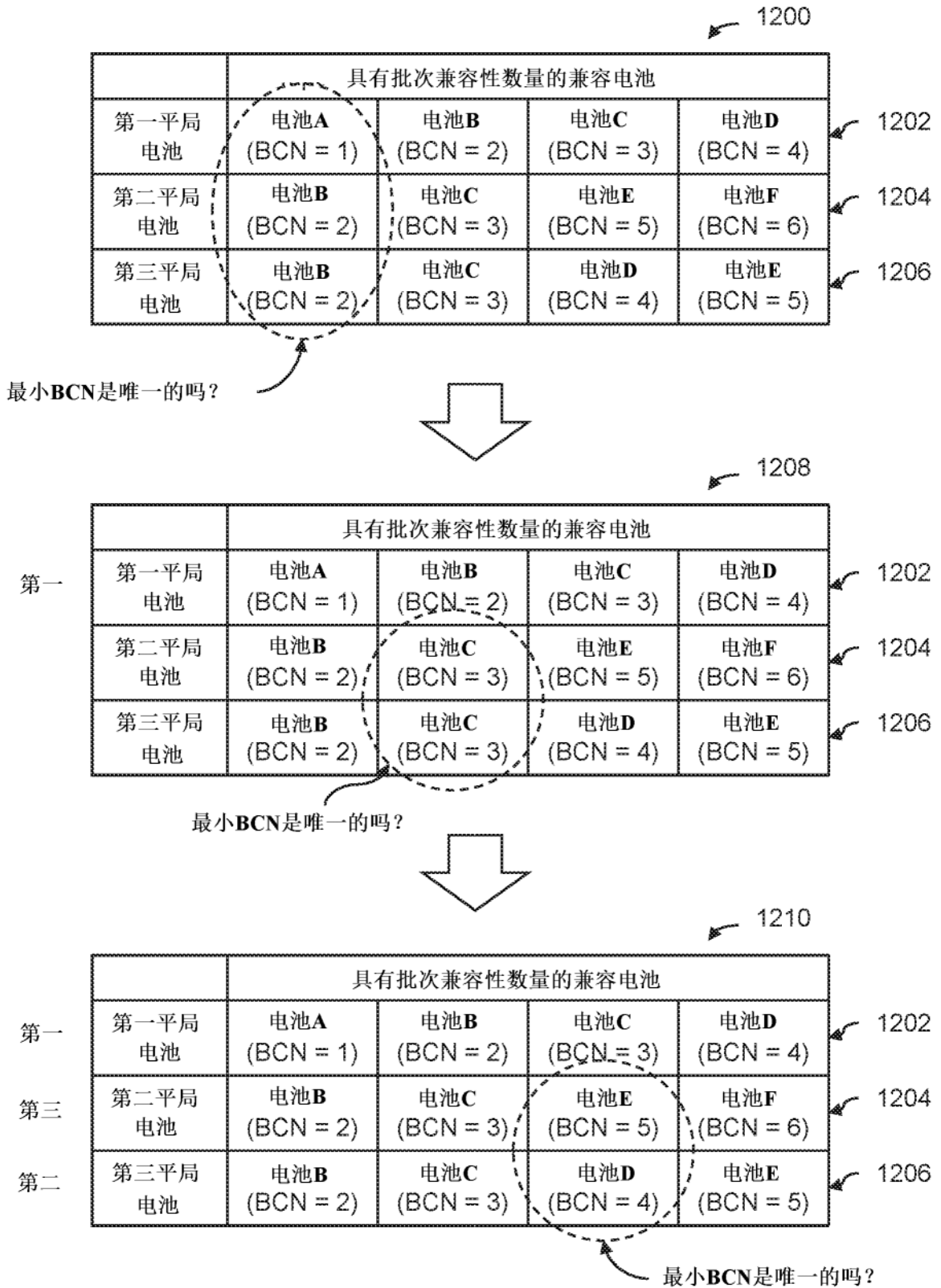


图 12



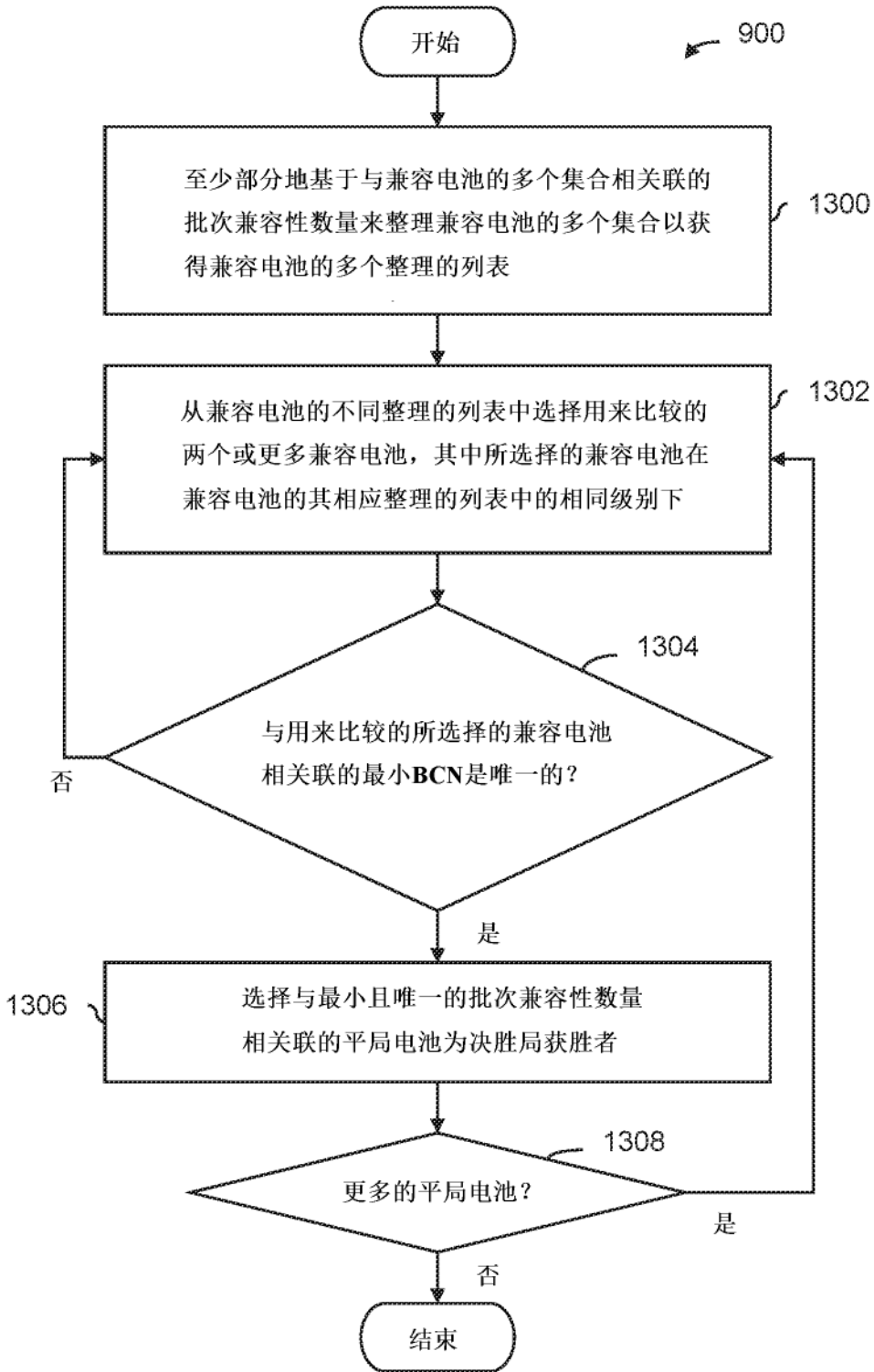


图 13