

(12) 按照专利合作条约所公布的国际申请

(19) 世界知识产权组织
国际局

(43) 国际公布日
2018 年 3 月 1 日 (01.03.2018)



(10) 国际公布号
WO 2018/036282 A1

- (51) 国际专利分类号:
G06F 9/50 (2006.01)
- (21) 国际申请号: PCT/CN2017/091802
- (22) 国际申请日: 2017 年 7 月 5 日 (05.07.2017)
- (25) 申请语言: 中文
- (26) 公布语言: 中文
- (30) 优先权:
201610715089.9 2016 年 8 月 24 日 (24.08.2016) CN
- (71) 申请人: 深圳市中兴微电子有限公司
(SANECHIPS TECHNOLOGY CO., LTD.) [CN/CN];
中国广东省深圳市南山区西丽街道留仙大道
中兴工业园, Guangdong 518055 (CN)。
- (72) 发明人: 李航 (LI, Hang); 中国广东省深圳市
南山区西丽街道留仙大道中兴工业园,
Guangdong 518055 (CN)。
- (74) 代理人: 北京派特恩知识产权代理有限公司 (CHINA PAT INTELLECTUAL PROPERTY OFFICE); 中国北京市海淀区海淀南路 21 号中关村知识产权大厦 B 座 2 层, Beijing 100080 (CN)。
- (81) 指定国 (除另有指明, 要求每一种可提供的国家保护): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL,

(54) Title: TASK SCHEDULING METHOD, DEVICE AND COMPUTER STORAGE MEDIUM

(54) 发明名称: 一种任务调度方法、装置及计算机存储介质

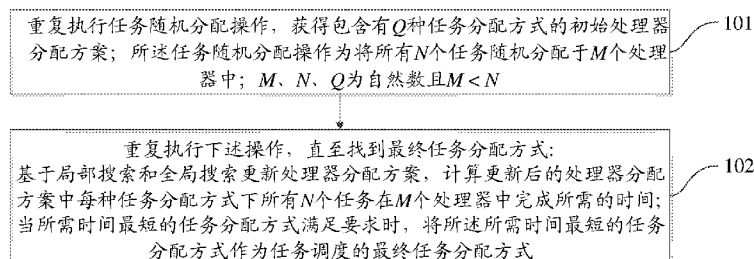


图 1

101 REPEATEDLY PERFORM A RANDOM ALLOCATION OPERATION FOR TASKS TO OBTAIN AN INITIAL PROCESSOR ALLOCATION SCHEME COMPRISING Q MANNERS OF TASK ALLOCATION, THE RANDOM ALLOCATION OPERATION FOR TASKS RANDOMLY ALLOCATES ALL OF N TASKS TO M PROCESSORS. WHEREIN M, N AND Q ARE NATURAL NUMBERS AND $M < N$

102 REPEATEDLY PERFORM THE FOLLOWING OPERATIONS UNTIL A FINAL MANNER OF TASK ALLOCATION IS FOUND: UPDATE THE PROCESSOR ALLOCATION SCHEME ON THE BASIS OF A LOCAL SEARCH AND A GLOBAL SEARCH, CALCULATE A TIME REQUIRED TO COMPLETE ALL OF N TASKS IN M PROCESSORS FROM EACH MANNER OF TASK ALLOCATION IN THE UPDATED PROCESSOR ALLOCATION SCHEME, AND IF A MANNER OF TASK ALLOCATION CONSUMING THE SHORTEST TIME SATISFIES A REQUIREMENT, USE THE MANNER OF TASK ALLOCATION CONSUMING THE SHORTEST TIME AS THE FINAL MANNER OF TASK ALLOCATION FOR TASK SCHEDULING

(57) Abstract: A task scheduling method, comprising: repeatedly performing a random allocation operation for tasks to obtain an initial processor allocation scheme comprising Q manners of task allocation, the random allocation operation for tasks randomly allocating all of N tasks to M processors, wherein M , N and Q are natural numbers and $M < N$ (101); repeatedly performing the following operations until a final manner of task allocation is found: updating the processor allocation scheme on the basis of a local search and a global search, calculating a time required to complete all of N tasks in M processors from each manner of task allocation in the updated processor allocation scheme, and if a manner of task allocation consuming the shortest time satisfies a requirement, using the manner of task allocation consuming the shortest time as the final manner of task allocation for task scheduling (102).

SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG,
US, UZ, VC, VN, ZA, ZM, ZW。

- (84) 指定国 (除另有指明, 要求每一种可提供的地区
保护): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ,
NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), 欧亚 (AM,
AZ, BY, KG, KZ, RU, TJ, TM), 欧洲 (AL, AT, BE, BG,
CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU,
IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT,
RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI,
CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG)。

本国际公布:

- 包括国际检索报告 (条约第21条(3))。

(57) 摘要: 一种任务调度方法, 包括: 重复执行任务随机分配操作, 获得包含有Q种任务分配方式的初始处理器分配方案; 所述任务随机分配操作为将所有N个任务随机分配于M个处理器中; M、N、Q为自然数且 $M < N$ (101); 重复执行下述操作, 直至找到最终任务分配方式: 基于局部搜索和全局搜索更新处理器分配方案, 计算更新后的处理器分配方案中每种任务分配方式下所有N个任务在M个处理器中完成所需的时间; 当所需时间最短的任务分配方式满足要求时, 将所述所需时间最短的任务分配方式作为任务调度的最终任务分配方式 (102)。

一种任务调度方法、装置及计算机存储介质

相关申请的交叉引用

本申请基于申请号为 201610715089.9、申请日为 2016 年 08 月 24 日的中国专利申请提出，并要求该中国专利申请的优先权，该中国专利申请的全部内容在此引入本申请作为参考。

技术领域

本发明涉及任务调度技术，尤其涉及一种任务调度方法、装置及计算机存储介质。

背景技术

在嵌入式多处理器系统中，经常会出现这样一种情况，即：一些处理器的负载很重，而另一些处理器的负载却很轻，甚至空闲。然而，这种情况无疑会降低整体系统的工作效率。为了提高处理器的利用率以及整体系统并行计算的效率，应该把负载过重的处理器上的一部分负载即任务转移到空闲的或负载较轻的处理器上，这就出现了关于处理器的负载分配问题的研究。

目前，解决处理器负载分配问题时，通常采用粒子群算法获得最优解作为处理器调度的方案；所谓传统粒子群算法是受鸟群觅食行为的启发而提出的，其基本思想是通过群体中个体之间的协作和信息共享来寻找最优解。粒子群算法将粒子群体中的每个个体看作是搜索空间中的一个没有体积的微粒，并且在搜索空间中从一个随机初始位置根据随机初始速度飞行。每个粒子代表解空间的一个候选解，粒子的飞行速度根据自身的飞行经验和同伴的飞行经验来进行动态调整。每个粒子在飞行过程中所经历过的最

好位置就是该粒子本身找到的个体最优解，整个粒子群体所经历过的最好位置就是整个粒子群体目前所找到的整体最优解。每个粒子都通过个体最优解和整体最优解不断地更新自己，从而产生新一代粒子群体，直到达到规定的迭代次数或者满足规定的误差标准为止。但是，现有的粒子群算法在求解调度问题时都存在早熟收敛、易陷入局部最优、进化后期算法收敛速度明显下降等缺点。

显然，现有技术中没有合适、优化的算法能用于解决求解调度问题，基于算法求解来实现处理器的负载分配和任务优化调度尚没有好的解决方案。

10 发明内容

有鉴于此，本发明实施例提供了一种任务调度方法、装置及计算机存储介质，能基于合适的算法求解来优化、全面、有效地解决任务调度问题。

为达到上述目的，本发明实施例的技术方案是这样实现的：

第一方面，本发明实施例提供了一种任务调度方法，所述方法包括：

15 重复执行任务随机分配操作，获得包含有 Q 种任务分配方式的初始处理器分配方案；所述任务随机分配操作为将所有 N 个任务随机分配于 M 个处理器中； M 、 N 、 Q 为自然数且 $M < N$ ；

重复执行下述操作，直至找到最终任务分配方式：

20 基于局部搜索和全局搜索更新处理器分配方案，计算更新后的处理器分配方案中每种任务分配方式下所有 N 个任务在 M 个处理器中完成所需的时间；

当所需时间最短的任务分配方式满足要求时，将所述所需时间最短的任务分配方式作为任务调度的最终任务分配方式。

在一实施例中，所述方法还包括：

25 当所需时间最短的任务分配方式不满足要求时，对所述更新后的处理

器分配方案中的任务分配方式执行变异操作。

在一实施例中，所述将所有 N 个任务随机分配于 M 个处理器中，包括：

随机选择 $[1, M]$ 之间的一个整数随机数 m_1 ，并从所有 N 个任务中随机选择一个任务分配给第 m_1 个处理器；随机选择 $[1, M]$ 之间的一个整数随机数 m_2 ，并从剩余的 $N-1$ 个任务中随机选择一个任务分配给第 m_2 个处理器；以此循环，直至所有 N 个任务都被分配给 M 个处理器，获得一种任务分配方式。

在一实施例中，所述基于局部搜索和全局搜索更新处理器分配方案，包括：

利用粒子群算法对当前的处理器分配方案进行局部搜索，获取所述当前的处理器分配方案中的局部最优任务分配方式；

将所述当前的处理器分配方案中的局部最优任务分配方式下所有 N 个任务在 M 个处理器中完成所需的时间作为第一时间；所述当前的处理器分配方案中的全局最优任务分配方式下所有 N 个任务在 M 个处理器中完成所需的时间作为第二时间；

确定所述第一时间小于或等于所述第二时间时，将所述当前的处理器分配方案中的全局最优任务分配方式替换为所述当前的处理器分配方案中的局部最优任务分配方式；

对所述当前的处理器分配方案中的任务分配方式执行交叉操作，获得更新后的处理器分配方案。

在一实施例中，所述利用粒子群算法对当前的处理器分配方案进行局部搜索，获取所述当前的处理器分配方案中的局部最优任务分配方式，包括：

随机生成一个包含有 I 个粒子且每个粒子为一种任务分配方式的初始粒子群，并将所述当前的处理器分配方案中的全局最优任务分配方式作为

所述初始粒子群的初始最优粒子加入到所述初始粒子群中；

根据 $V_i(k+1) = V_i(k) \otimes X_{gbest}(k) \otimes X_{ibest}(k)$ 更新在第 $k+1$ 次迭代时粒子 i 的速度 $V_i(k+1)$, $1 \leq i \leq I+1$, $0 \leq k \leq K-1$; 其中, $X_{gbest}(k)$ 表示在第 $k+1$ 次迭代之前所述粒子群中最优粒子的位置, $X_{ibest}(k)$ 表示在第 $k+1$ 次迭代之前粒子 i 的最优位置, $\otimes$ 表示交叉操作, K 为设置的最大迭代次数; 所述最优粒子为粒子群中具有最小适应度值的粒子; 所述适应度值为粒子在所对应的任务分配方式下所有 N 个任务在 M 个处理器中完成所需的时间;

根据 $X_i(k+1) = X_i(k) \otimes V_i(k+1)$ 更新在第 $k+1$ 次迭代时粒子 i 的位置 $X_i(k+1)$;

10 计算粒子 i 在第 $k+1$ 次迭代时的适应度值, 并根据粒子 i 在第 $k+1$ 次迭代时的适应度值更新所述粒子 i 在第 $k+1$ 次迭代后的最优位置 $X_{ibest}(k+1)$ 、以及更新在第 $k+1$ 次迭代后所述粒子群中的最优粒子和所述粒子群中更新后的最优粒子的位置 $X_{gbest}(k+1)$;

15 当检测出所述迭代次数 $k+1$ 小于 K 且所述粒子群中最优粒子的适应度值小于设置的第一阈值或所述迭代次数 $k+1$ 等于 K 时, 将所述粒子群中的最优粒子所对应的任务分配方式作为所述当前的处理器分配方案中的局部最优任务分配方式;

20 当检测出所述迭代次数 $k+1$ 小于 K 且所述粒子群中最优粒子的适应度值大于或等于设置的第一阈值时, 计算第 $k+1$ 次迭代后所述粒子群中粒子之间的相似度值, 若所述相似度值大于或等于设置的第二阈值, 则对所述粒子的位置执行变异操作。

在一实施例中, 所述随机生成一个包含有 I 个粒子且每个粒子为一种任务分配方式的初始粒子群, 包括:

25 从所有 N 个任务中随机选择一个任务分配给第一个处理器; 从剩余的 $N-1$ 个任务中随机选择一个任务分配给所述第二个处理器; 从剩余的 $N-m$

个任务中随机选择一个任务分配给所述第 m 个处理器, $3 \leq m \leq M$; 以此循环, 直至所有 N 个任务都被分配给 M 个处理器, 获得一个对应一种任务分配方式的粒子; 其中, 当 m 大于 M 即随机分配了 M 个任务后, 重新自第一个处理器开始分配;

- 5 循环执行上述分配过程, 直至获取到与 I 种任务分配方式一一对应的 I 个粒子。

第二方面, 本发明实施例提供了一种任务调度装置, 所述装置包括: 初始化模块、处理模块; 其中,

- 所述初始化模块, 配置为重复执行任务随机分配操作, 获得包含有 Q 种
10 任务分配方式的初始处理器分配方案; 所述任务随机分配操作为将所有 N 个任务随机分配于 M 个处理器中; M 、 N 、 Q 为自然数且 $M < N$;

所述处理模块, 配置为重复执行下述操作, 直至找到最终任务分配方式:

- 基于局部搜索和全局搜索更新处理器分配方案, 计算更新后的处理器
15 分配方案中每种任务分配方式下所有 N 个任务在 M 个处理器中完成所需的时间;

当所需时间最短的任务分配方式满足要求时, 将所述所需时间最短的任务分配方式作为任务调度的最终任务分配方式。

- 在一实施例中, 所述处理模块, 还配置为当所需时间最短的任务分配
20 方式不满足要求时, 对所述更新后的处理器分配方案中的任务分配方式执行变异操作。

在一实施例中, 所述初始化模块将所有 N 个任务随机分配于 M 个处理器中为:

- 随机选择 $[1, M]$ 之间的一个整数随机数 m_1 , 并从所有 N 个任务中随机选
25 择一个任务分配给第 m_1 个处理器; 随机选择 $[1, M]$ 之间的一个整数随机数

m_2 ，并从剩余的 $N-1$ 个任务中随机选择一个任务分配给第 m_2 个处理器；以此循环，直至所有 N 个任务都被分配给 M 个处理器，获得一种任务分配方式。

在一实施例中，所述处理模块基于局部搜索和全局搜索更新处理器分配方案为：

利用粒子群算法对当前的处理器分配方案进行局部搜索，获取所述当前的处理器分配方案中的局部最优任务分配方式；

将所述当前的处理器分配方案中的局部最优任务分配方式下所有 N 个任务在 M 个处理器中完成所需的时间作为第一时间；所述当前的处理器分配方案中的全局最优任务分配方式下所有 N 个任务在 M 个处理器中完成所需的时间作为第二时间；

确定所述第一时间小于或等于所述第二时间时，将所述当前的处理器分配方案中的全局最优任务分配方式替换为所述当前的处理器分配方案中的局部最优任务分配方式；

对所述当前的处理器分配方案中的任务分配方式执行交叉操作，获得更新后的处理器分配方案。

在一实施例中，所述处理模块利用粒子群算法对当前的处理器分配方案进行局部搜索，获取所述当前的处理器分配方案中的局部最优任务分配方式：

随机生成一个包含有 I 个粒子且每个粒子为一种任务分配方式的初始粒子群，并将所述当前的处理器分配方案中的全局最优任务分配方式作为所述初始粒子群的初始最优粒子加入到所述初始粒子群中；

根据 $V_i(k+1) = V_i(k) \otimes X_{gbest}(k) \otimes X_{ibest}(k)$ 更新在第 $k+1$ 次迭代时粒子 i 的速度 $V_i(k+1)$ ， $1 \leq i \leq I+1$ ， $0 \leq k \leq K-1$ ；其中， $X_{gbest}(k)$ 表示在第 $k+1$ 次迭代之前所述粒子群中最优粒子的位置， $X_{ibest}(k)$ 表示在第 $k+1$ 次迭代之前粒子 i 的最

优位置, $\otimes$ 表示交叉操作, K 为设置的最大迭代次数; 所述最优粒子为粒子群中具有最小适应度值的粒子; 所述适应度值为粒子在所对应的任务分配方式下所有 N 个任务在 M 个处理器中完成所需的时间;

根据 $X_i(k+1) = X_i(k) \otimes V_i(k+1)$ 更新在第 $k+1$ 次迭代时粒子 i 的位置

5 $X_i(k+1)$;

计算粒子 i 在第 $k+1$ 次迭代时的适应度值, 并根据粒子 i 在第 $k+1$ 次迭代时的适应度值更新所述粒子 i 在第 $k+1$ 次迭代后的最优位置 $X_{ibest}(k+1)$ 、以及更新在第 $k+1$ 次迭代后所述粒子群中的最优粒子和所述粒子群中更新后的最优粒子的位置 $X_{gbest}(k+1)$;

10 当检测出所述迭代次数 $k+1$ 小于 K 且所述粒子群中最优粒子的适应度值小于设置的第一阈值或所述迭代次数 $k+1$ 等于 K 时, 将所述粒子群中的最优粒子所对应的任务分配方式作为所述当前的处理器分配方案中的局部最优任务分配方式;

15 当检测出所述迭代次数 $k+1$ 小于 K 且所述粒子群中最优粒子的适应度值大于或等于设置的第一阈值时, 计算第 $k+1$ 次迭代后所述粒子群中粒子之间的相似度值, 若所述相似度值大于或等于设置的第二阈值, 则对所述粒子的位置执行变异操作。

在一实施例中, 所述处理模块随机生成一个包含有 I 个粒子且每个粒子为一种任务分配方式的初始粒子群为:

20 从所有 N 个任务中随机选择一个任务分配给第一个处理器; 从剩余的 $N-1$ 个任务中随机选择一个任务分配给所述第二个处理器; 从剩余的 $N-m$ 个任务中随机选择一个任务分配给所述第 m 个处理器, $3 \leq m \leq M$; 以此循环, 直至所有 N 个任务都被分配给 M 个处理器, 获得一个对应一种任务分配方式的粒子; 其中, 当 m 大于 M 即随机分配了 M 个任务后, 重新自第一个处
25 理器开始分配;

循环执行上述分配过程，直至获取到与 I 种任务分配方式一一对应的 I 个粒子。

第三方面，本发明实施例还提供了一种计算机存储介质，所述计算机存储介质中存储有计算机可执行指令，所述计算机可执行指令用于执行本
5 发明实施例所述的任务调度方法。

本发明实施例提供的任务调度方法、装置及计算机存储介质，重复执行任务随机分配操作，获得包含有 Q 种任务分配方式的初始处理器分配方案；所述任务随机分配操作为将所有 N 个任务随机分配于 M 个处理器中； M 、 N 、 Q 为自然数且 $M < N$ ；重复执行下述操作，直至找到最终任务分配
10 方式：基于局部搜索和全局搜索更新处理器分配方案，计算更新后的处理器分配方案中每种任务分配方式下所有 N 个任务在 M 个处理器中完成所需的时间；当所需时间最短的任务分配方式满足要求时，将所述所需时间最短的任务分配方式作为任务调度的最终任务分配方式。可见，本发明实施例通过将在全局空间搜索全局最优解和在局部空间搜索局部最优解结合起来，两者相互协调和促进，以求取最优的任务分配方式。
15

此外，利用遗传算法求解任务调度的全局最优任务分配方式，而利用粒子群算法求解任务调度的局部最优任务分配方式，并将局部最优任务分配方式作为全局最优任务分配方式进化的依据，能够在粒子群整体陷入局部最优时，通过变异操作改变粒子的当前位置和速度，解决了粒子群算法
20 在求解调度问题时容易出现早熟收敛、易陷入局部最优等缺陷。

附图说明

图 1 为本发明实施例任务调度方法的实现流程示意图；

图 2 为本发明实施例任务调度方法的具体实现流程示意图；

图 3 为本发明实施例中所述利用粒子群算法对染色体种群进行寻优的
25 具体实现流程示意图；

图 4 为本发明实施例任务调度装置的组成结构示意图。

具体实施方式

下面将结合本发明实施例中的附图，对本发明实施例中的技术方案进行清楚、完整地描述。

5 遗传算法是模拟达尔文生物进化论的自然选择和遗传学机理的生物进化过程的计算模型，是一种通过模拟自然进化过程搜索最优解的方法。该方法的处理过程是这样的：首先，要创建一个随机的初始状态即一群初始解，将这些解比喻为染色体或基因，可将该种群称为第一代；接着，根据问题求解的实际接近程度对每一个染色体指定一个合理的适应度值；然后，
10 对染色体进行各种进化操作如选择、交叉和变异等。在以上的各种进化操作过程中，通过选择适当的控制参数可以使算法获得令人满意的结果。现有的研究已证明，遗传算法具有自学习性和自适应性，并能够减少陷入局部最优的风险。

 本发明实施例正是利用遗传算法的优势，将遗传算法求解用于完成处
15 理器负载分配和任务调度。

 图 1 为本发明实施例任务调度方法的实现流程示意图，该方法包括：

 步骤 101：重复执行任务随机分配操作，获得包含有 Q 种任务分配方式的初始处理器分配方案；所述任务随机分配操作为将所有 N 个任务随机分配于 M 个处理器中；

20 具体地，重复执行 Q 次任务随机分配操作，获得包含有 Q 种任务分配方式的初始处理器分配方案，即每执行一次任务随机分配操作，则获得一种所有 N 个任务被分配于 M 个处理器中的任务分配方式；所述任务随机分配操作为将所有 N 个任务随机分配于 M 个处理器中。

 这里，所述将所有 N 个任务随机分配于 M 个处理器中，包括：随机选
25 择 $[1, M]$ 之间的一个整数随机数 m_i ，并从所有 N 个任务中随机选择一个任务

分配给第 m_1 个处理器；随机选择 $[1, M]$ 之间的一个整数随机数 m_2 ，并从剩余的 $N-1$ 个任务中随机选择一个任务分配给第 m_2 个处理器；以此循环，直至所有 N 个任务都被分配给 M 个处理器，以获得一种所有 N 个任务被分配于 M 个处理器中的任务分配方式。

- 5 这里，所述 M 、 N 、 Q 为自然数且 $M < N$ ；任务之间都是独立的，每个任务所具有的负载可能相同或不相同，处理器之间也都是独立的，负载的大小影响处理器对任务的完成时间；在实际应用中，为保证任务分配方式之间的差异性而提高泛化性， Q 可设置为大于或等于 M ；由于受到随机因素和 Q 的大小的影响，所述初始处理器分配方案中包含的 Q 种任务分配方式
- 10 中可能存在相同的任务分配方式，因此，可将相同的任务分配方式从所述初始处理器分配方案中删除，以提高所述初始处理器分配方案中任务分配方式的多样性。

- 步骤 102：重复执行下述操作，直至找到最终任务分配方式：基于局部搜索和全局搜索更新处理器分配方案，计算更新后的处理器分配方案中每
- 15 种任务分配方式下所有 N 个任务在 M 个处理器中完成所需的时间；当所需时间最短的任务分配方式满足要求时，将所述所需时间最短的任务分配方式作为任务调度的最终任务分配方式。

- 这里，所述所需时间最短的任务分配方式满足要求是指所述所需时间最短的任务分配方式下所有 N 个任务在 M 个处理器中完成所需的时间与所有
- 20 有 N 个任务在 M 个处理器中完成所需的理想时间之差小于或等于预设时间阈值；所述所有 N 个任务在 M 个处理器中完成所需的理想时间为所有 N 个任务在任意一个处理器中完成所需的时间 T 除以 M ，即所述理想时间可表示为 T/M 。

- 这里，所述基于局部搜索和全局搜索更新处理器分配方案，包括：
- 25 利用粒子群算法对当前的处理器分配方案进行局部搜索，获取所述当

前的处理器分配方案中的局部最优任务分配方式;

将所述当前的处理器分配方案中的局部最优任务分配方式下所有 N 个任务在 M 个处理器中完成所需的时间作为第一时间; 所述当前的处理器分配方案中的全局最优任务分配方式下所有 N 个任务在 M 个处理器中完成所需的时间作为第二时间;

确定所述第一时间小于或等于所述第二时间时, 将所述当前的处理器分配方案中的全局最优任务分配方式替换为所述当前的处理器分配方案中的局部最优任务分配方式;

对所述当前的处理器分配方案中的任务分配方式执行交叉操作, 获得更新后的处理器分配方案。

这里, 所述当前的处理器分配方案可能是步骤 101 中生成的初始处理器分配方案, 也可能是对步骤 101 中生成的初始处理器分配方案进行一次或一次以上更新后形成的处理器分配方案, 但所述当前的处理器分配方案中都包含有 Q 种任务分配方式; 此外, 对所述当前的处理器分配方案中的任务分配方式执行交叉操作可以是对全部 Q 种任务分配方式中的两种或两种以上任务分配方式按照遗传算法执行交叉操作, 且交叉操作的次数可以是一次或一次以上; 当两种任务分配方式执行完一次以上交叉操作后, 则将经交叉后生成的所有新的任务分配方式中所需时间最短的两种任务分配方式作为该两种任务分配方式进行更新后生成的任务分配方式; 当 Q 种任务分配方式中的每两种任务分配方式都执行完一次以上的交叉操作后, 将会生成数量大于 Q 的新的任务分配方式, 通过计算每一种新的任务分配方式下所有 N 个任务在 M 个处理器中完成所需的时间, 将所需时间最短的 Q 种新的任务分配方式作为更新后的处理器分配方案, 以保证更新后获得的处理器分配方案中包含 Q 种任务分配方式。

这里, 通过对任意两种任务分配方式执行交叉操作, 使不同任务分配

方式中的部分任务进行互换即互换分配给不同处理器的任务，以生成新的任务分配方式，从而完成对处理器分配方案的更新处理，能够避免陷入早熟收敛。

这里，所述利用粒子群算法对当前的处理器分配方案进行局部搜索，

5 获取所述当前的处理器分配方案中的局部最优任务分配方式，包括：

随机生成一个包含有 I 个粒子且每个粒子为一种任务分配方式的初始粒子群，并将所述当前的处理器分配方案中的全局最优任务分配方式作为所述初始粒子群的初始最优粒子加入到所述初始粒子群中；

10 根据 $V_i(k+1) = V_i(k) \otimes X_{gbest}(k) \otimes X_{ibest}(k)$ 更新在第 $k+1$ 次迭代时粒子 i 的速度 $V_i(k+1)$ ， $1 \leq i \leq I+1$ ， $0 \leq k \leq K-1$ ；其中， $X_{gbest}(k)$ 表示在第 $k+1$ 次迭代之前所述粒子群中最优粒子的位置， $X_{ibest}(k)$ 表示在第 $k+1$ 次迭代之前粒子 i 的最优位置， $\otimes$ 表示交叉操作， K 为设置的最大迭代次数；所述最优粒子为粒子群中具有最小适应度值的粒子；所述适应度值为粒子在所对应的任务分配方式下所有 N 个任务在 M 个处理器中完成所需的时间；

15 根据 $X_i(k+1) = X_i(k) \otimes V_i(k+1)$ 更新在第 $k+1$ 次迭代时粒子 i 的位置 $X_i(k+1)$ ；

20 计算粒子 i 在第 $k+1$ 次迭代时的适应度值，并根据粒子 i 在第 $k+1$ 次迭代时的适应度值更新所述粒子 i 在第 $k+1$ 次迭代后的最优位置 $X_{ibest}(k+1)$ 、以及更新在第 $k+1$ 次迭代后所述粒子群中的最优粒子和所述粒子群中更新后的最优粒子的位置 $X_{gbest}(k+1)$ ；

当检测出所述迭代次数 $k+1$ 小于 K 且所述粒子群中最优粒子的适应度值小于设置的第一阈值或所述迭代次数 $k+1$ 等于 K 时，将所述粒子群中的最优粒子所对应的任务分配方式作为所述当前的处理器分配方案中的局部最优任务分配方式；

25 当检测出所述迭代次数 $k+1$ 小于 K 且所述粒子群中最优粒子的适应度

值大于或等于设置的第一阈值时,计算第 $k+1$ 次迭代后所述粒子群中粒子之间的相似度值,若所述相似度值大于或等于设置的第二阈值,则对所述粒子的位置执行变异操作。

这里,除了将所述当前的处理器分配方案中的全局最优任务分配方式加入到所述初始粒子群中以外,还可以将所述当前的处理器分配方案中所需时间较少的多种任务分配方式都加入到所述初始粒子群中,并以所述当前的处理器分配方案中的全局最优任务分配方式作为所述初始粒子群的初始最优粒子;例如,可将所述当前的处理器分配方案中的前5种所需时间最短的任务分配方式加入所述初始粒子群中。

这里,当 $k=0$ 时, $X_{gbest}(0)$ 表示所述初始粒子群中最优粒子的位置,即所述初始粒子群中最优粒子对应的任务分配方式; $X_{ibest}(0)$ 表示所述初始粒子群中粒子 i 的初始最优位置,即所述初始粒子群中粒子 i 对应的任务分配方式; $V_i(0)$ 表示所述初始粒子群中粒子 i 的初始速度,为 $(T_i(0)*M*v)/T_1(0)$;其中, $T_i(0)$ 表示所述初始粒子群中粒子 i 对应的任务分配方式下所有 N 个任务在 M 个处理器中完成所需的时间, $T_1(0)$ 表示所述初始粒子群中最优粒子对应的任务分配方式下所有 N 个任务在 M 个处理器中完成所需的时间, v 表示速度参数。

这里,当对所述当前的粒子群中的粒子的速度和位置进行更新后,粒子对应的任务分配方式中给每个处理器分配的任务数量发生变化,所述当前的粒子群中的粒子都向所述当前的粒子群中的最优粒子靠近,即粒子对应的任务分配方式都朝向最优粒子对应的任务分配方式靠近,以使粒子对应的任务分配方式中每个处理器的任务负载趋向于相等。

这里,当第 $k+1$ 次迭代后所生成的粒子群中粒子 i 的适应度值小于第 k 次迭代后所生成的粒子群中粒子 i 的适应度值时,则保持粒子 i 在第 $k+1$ 次迭代后的最优位置 $X_{ibest}(k+1)$ 为粒子 i 在第 k 次迭代后的最优位置 $X_{ibest}(k)$;当第

$k+1$ 次迭代后所生成的粒子群中最优粒子的适应度值小于第 k 次迭代后所生成的粒子群中最优粒子的适应度值时,则第 $k+1$ 次迭代后所生成的粒子群中最优粒子仍然保持为第 k 次迭代后所生成的粒子群中最优粒子、第 $k+1$ 次迭代后所生成的粒子群中最优粒子的位置 $X_{gbest}(k+1)$ 仍然保持为第 k 次迭代后所生成的粒子群中最优粒子的位置 $X_{gbest}(k)$ 。

这里,根据公式 $similarity(X_{ij}) = \sum_{f=1}^M same(X_i^f, X_j^f)$, $same(x, y) = \begin{cases} 0, & \text{if } x = y \\ 1, & \text{if } x \neq y \end{cases}$ 计

算所述粒子的位置相互之间的相似度值 $similarity(X_{ij})$; 其中, X_i^f 、 X_j^f 分别代表粒子 i 和粒子 j 对应的任务分配方式中相同处理器被分配的任务。

这里,可以计算所有 $I+1$ 个粒子中两个或两个以上粒子之间的相似度值,比如,当计算出两个粒子之间的相似度值大于设置的第二阈值时,可认为是这两个粒子分别所对应的任务分配方式中将相同的任务分配给了相同的处理器;若两个粒子之间的相似度值大于设置的第二阈值时,可对这两个粒子分别执行变异操作,以使粒子所对应的任务分配方式中分配给不同处理器的任务进行互换,从而避免陷入局部最优解。

这里,可以只对所有 $I+1$ 个粒子中两个或两个以上粒子执行变异操作,而不必对所有 $I+1$ 个粒子执行变异操作。例如,当有三个粒子且每两个粒子相互之间的相似度值都大于设置的第二阈值时,可只对所述三个粒子中的任意两个粒子执行变异操作。

这里,所述随机生成一个包含有 I 个粒子且每个粒子为一种任务分配方式的初始粒子群,包括:

从所有 N 个任务中随机选择一个任务分配给第一个处理器;从剩余的 $N-1$ 个任务中随机选择一个任务分配给所述第二个处理器;从剩余的 $N-m$ 个任务中随机选择一个任务分配给所述第 m 个处理器, $3 \leq m \leq M$;以此循环,直至所有 N 个任务都被分配给 M 个处理器,获得一个对应一种任务分配方

式的粒子；其中，当 m 大于 M 即随机分配了 M 个任务后，重新自第一个处理器开始分配；

循环执行上述分配过程，直至获取到与 I 种任务分配方式一一对应的 I 个粒子。

5 这里，当 m 大于 M 时，即已将 m 个任务随机分配给 M 个处理器，而每个处理器被分配了一个任务后，重新自第一个处理器开始分配任务，即从剩余的 $N-M$ 个任务中随机选择一个任务分配给所述第一个处理器，从剩余的 $N-M-1$ 个任务中随机选择一个任务分配给所述第二个处理器，并以此类推。

10 进一步地，该方法还包括：当所需时间最短的任务分配方式不满足要求时，对所述更新后的处理器分配方案中的任务分配方式执行变异操作。

 作为一种实施方式，当步骤 102 中获取的所需时间最短的任务分配方式不满足要求时，即所述所需时间最短的任务分配方式下所有 N 个任务在 M 个处理器中完成所需的时间与所有 N 个任务在 M 个处理器中完成所需的理想时间之差大于预设时间阈值，则对所述更新后的处理器分配方案中的任务分配方式执行变异操作，然后继续执行步骤 102。

 这里，所述对所述更新后的处理器分配方案中的任务分配方式执行变异操作的次数可以是对全部 Q 种任务分配方式中的一种或一种以上任务分配方式按照遗传算法执行变异操作，且变异操作的次数可以是一次或一次以上；当一种任务分配方式执行完一次以上变异操作后，则将经变异后生成的所有新的任务分配方式中所需时间最短的任务分配方式作为该任务分配方式进行更新后的任务分配方式；当 Q 种任务分配方式中的每一种任务分配方式都执行完一次以上变异操作后，将会生成数量大于 Q 的新的任务分配方式，通过计算每一个新的任务分配方式下所有 N 个任务在 M 个处理器中完成所需的时间，将所需时间最短的 Q 种新的任务分配方式作为更新后的任

务分配方式，以保证更新后获得的处理器分配方案中包含 Q 种任务分配方式。

这里，当所需时间最短的任务分配方式不满足要求即粒子群整体陷入局部最优时，通过对任意任务分配方式执行变异操作，改变粒子的位置
5 速度，从而使粒子群整体跳出局部最优继续寻优。

下面以一个具体的实施例对所述任务调度方法进行举例说明，图 2 为所述任务调度方法的具体实现流程示意图，该方法包括：

步骤 201：随机生成初始染色体种群；

作为一种实施方式，重复执行 Q 次任务随机分配操作，获得包含有 Q 条
10 染色体的初始染色体种群；其中，所述任务随机分配操作为将所有 N 个任务随机分配于 M 个处理器中。

这里，所述将所有 N 个任务随机分配于 M 个处理器中，包括：随机选择 $[1, M]$ 之间的一个整数随机数 m_1 ，并从所有 N 个任务中随机选择一个任务分配给第 m_1 个处理器；随机选择 $[1, M]$ 之间的一个整数随机数 m_2 ，并从剩余的
15 的 $N-1$ 个任务中随机选择一个任务分配给第 m_2 个处理器；以此循环，直至所有 N 个任务都被分配给 M 个处理器，获得一种所有 N 个任务被分配于 M 个处理器中的任务分配方式即一条染色体。

这里，每条染色体代表一种任务分配方式；所述 M 、 N 、 Q 为自然数且 $M < N$ ；任务之间都是独立的，每个任务所具有的负载可能相同或不相同，
20 处理器之间也都是独立的，负载的大小影响处理器对任务的完成时间；在实际应用中，为保证任务分配方式之间的差异性而提高泛化性， Q 可设置为大于或等于 M ；由于受到随机因素和 Q 的大小的影响，所述初始染色体种群中可能存在相同的染色体即可能存在相同的任务分配方式，因此，可将相同的染色体从所述初始染色体种群中删除，以提高所述初始染色体种群
25 中染色体的多样性。

步骤 202: 利用粒子群算法对染色体种群进行寻优, 获取粒子群的最优解;

作为一种实施方式, 利用粒子群算法对当前的染色体种群进行局部寻优搜索, 获取粒子群的最优解即所述当前的染色体种群的局部最优任务分配方式。

这里, 所述当前的染色体种群可能是步骤 201 中生成的初始染色体种群, 也可能是对步骤 201 中生成的初始染色体种群进行一次或一次以上更新后生成的染色体种群, 但所述当前的染色体种群中都包含有 Q 条染色体。

图 3 为所述利用粒子群算法对染色体种群进行寻优的具体实现流程图, 该寻优过程包括:

步骤 20201: 随机生成初始粒子群;

作为一种实施方式, 随机生成一个包含有 I 个粒子且每个粒子为一种任务分配方式的初始粒子群, 并将所述当前的染色体种群中的最优染色体作为所述初始粒子群的初始最优粒子加入到所述初始粒子群中, 以形成一个包含有 $I+1$ 个粒子的初始粒子群。

这里, 所述当前的染色体种群中的最优染色体为所有 N 个任务在 M 个处理器中完成所需的时间最短的任务分配方式所对应的染色体; 除了将所述当前的染色体种群中的最优染色体加入到所述初始粒子群中以外, 还可以将所述当前的染色体种群中所需时间较少的多个染色体都加入到所述初始粒子群中, 并以所述当前的染色体种群中的最优染色体作为所述初始粒子群的初始最优粒子; 例如, 可将所述当前的染色体种群中的前 5 个所需时间最短的任务分配方式所对应的染色体加入所述初始粒子群中。

这里, 所述随机生成一个包含有 I 个粒子且每个粒子为一种任务分配方式的初始粒子群, 包括:

从所有 N 个任务中随机选择一个任务分配给第一个处理器; 从剩余的

$N-1$ 个任务中随机选择一个任务分配给所述第二个处理器；从剩余的 $N-m$ 个任务中随机选择一个任务分配给所述第 m 个处理器， $3 \leq m \leq M$ ；以此循环，直至所有 N 个任务都被分配给 M 个处理器，获得一个对应一种任务分配方式的粒子；其中，当 m 大于 M 即随机分配了 M 个任务后，重新自第一个处理器开始分配；

循环执行上述分配过程，直至获取到与 I 种任务分配方式一一对应的 I 个粒子。

这里，当 m 大于 M 时，即已将 m 个任务随机分配给 M 个处理器，而每个处理器被分配了一个任务后，重新自第一个处理器开始分配任务，即从剩余的 $N-M$ 个任务中随机选择一个任务分配给所述第一个处理器，从剩余的 $N-M-1$ 个任务中随机选择一个任务分配给所述第二个处理器，并以此类推。

步骤 20202：更新粒子群中粒子的速度和位置；

作为一种实施方式，根据 $V_i(k+1) = V_i(k) \otimes X_{gbest}(k) \otimes X_{ibest}(k)$ 更新在第 $k+1$ 次迭代时当前的粒子群中粒子 i 的速度 $V_i(k+1)$ ， $1 \leq i \leq I+1$ ， $0 \leq k \leq K-1$ ；其中， $X_{gbest}(k)$ 表示在第 $k+1$ 次迭代之前当前的粒子群中最优粒子的位置， $X_{ibest}(k)$ 表示在第 $k+1$ 次迭代之前当前的粒子群中粒子 i 的最优位置， $\otimes$ 表示交叉操作， K 为设置的最大迭代次数；所述最优粒子为粒子群中具有最小适应度值的粒子；所述适应度值为粒子在所对应的任务分配方式下所有 N 个任务在 M 个处理器中完成所需的时间；根据 $X_i(k+1) = X_i(k) \otimes V_i(k+1)$ 更新在第 $k+1$ 次迭代时当前的粒子群中粒子 i 的位置 $X_i(k+1)$ 。

这里，当 $k=0$ 时， $X_{gbest}(0)$ 表示所述初始粒子群中最优粒子的位置，即所述初始粒子群中最优粒子对应的任务分配方式； $X_{ibest}(0)$ 表示所述初始粒子群中粒子 i 的初始最优位置，即所述初始粒子群中粒子 i 对应的任务分配方式； $V_i(0)$ 表示所述初始粒子群中粒子 i 的初始速度，为 $(T_i(0) * M * v) / T_i(0)$ ；

其中, $T_i(0)$ 表示所述初始粒子群中粒子 i 对应的任务分配方式下所有 N 个任务在 M 个处理器中完成所需的时间, $T_1(0)$ 表示所述初始粒子群中最优粒子对应的任务分配方式下所有 N 个任务在 M 个处理器中完成所需的时间, v 表示速度参数。

5 这里, 当对所述当前的粒子群中的粒子的速度和位置进行更新后, 粒子对应的任务分配方式中给每个处理器分配的任务数量发生变化, 所述当前的粒子群中的粒子都向所述当前的粒子群中的最优粒子靠近, 即粒子对应的任务分配方式都朝向最优粒子对应的任务分配方式靠近, 以使粒子对应的任务分配方式中每个处理器的任务负载趋向于相等。

10 步骤 20203: 更新粒子群中粒子的最优位置以及最优粒子;

作为一种实施方式, 根据步骤 20202 中所述当前的粒子群中每个粒子更新后获得的速度和位置, 更新所述当前的粒子群中每个粒子的最优位置, 更新所述当前的粒子群中的最优粒子以及所述当前的粒子群中的最优粒子的位置。

15 这里, 所述最优粒子为粒子群中具有最小适应度值的粒子; 所述适应度值为粒子在所对应的任务分配方式下所有 N 个任务在 M 个处理器中完成所需的时间; 当第 $k+1$ 次迭代后所生成的粒子群中粒子 i 的适应度值小于第 k 次迭代后所生成的粒子群中粒子 i 的适应度值时, 则保持粒子 i 在第 $k+1$ 次迭代后的最优位置 $X_{ibest}(k+1)$ 为粒子 i 在第 k 次迭代后的最优位置 $X_{ibest}(k)$; 当
20 第 $k+1$ 次迭代后所生成的粒子群中最优粒子的适应度值小于第 k 次迭代后所生成的粒子群中最优粒子的适应度值时, 则第 $k+1$ 次迭代后所生成的粒子群中最优粒子仍然保持为第 k 次迭代后所生成的粒子群中最优粒子、第 $k+1$ 次迭代后所生成的粒子群中最优粒子的位置 $X_{gbest}(k+1)$ 仍然保持为第 k 次迭代后所生成的粒子群中最优粒子的位置 $X_{gbest}(k)$ 。

25 步骤 20204: 判断粒子群的迭代次数是否小于最大迭代次数, 若小于,

则继续执行步骤 20205, 否则执行步骤 20208;

作为一种实施方式, 判断步骤 20202 中所述当前的粒子群的迭代次数 $k+1$ 的值是否小于设置的最大迭代次数 K , 若小于, 则继续执行步骤 20205; 否则, 判定所述对染色体种群进行局部寻优的过程结束并执行步骤 20208。

- 5 步骤 20205: 判断粒子群的最优解的适应度值是否小于设置的第一阈值, 若满足, 则执行步骤 20208, 否则继续执行步骤 20206;

作为一种实施方式, 判断步骤 20203 中获得的所述当前的粒子群中的最优粒子的适应度值是否小于设置的第一阈值, 若小于, 则判定对染色体种群进行局部寻优的过程结束并执行步骤 20208, 否则继续执行步骤 20206。

- 10 步骤 20206: 判断粒子群中粒子的相似度值是否小于设置的第二阈值, 若小于, 则返回执行步骤 20202, 否则继续执行步骤 20207;

作为一种实施方式, 计算粒子群中每两个粒子之间的相似度值, 当所述相似度值大于或等于设置的第二阈值时, 继续执行步骤 20207; 当所述相似度值小于设置的第二阈值时, 返回执行步骤 20202。

- 15 这里, 可以计算所有 $I+1$ 个粒子中两个或两个以上粒子之间的相似度值; 例如, 当计算出两个粒子之间的相似度值大于设置的第二阈值时, 可认为是这两个粒子分别所对应的任务分配方式中将相同的任务分配给了相同的处理器。

步骤 20207: 对粒子群中的粒子执行变异操作, 并返回执行步骤 20202;

- 20 作为一种实施方式, 当步骤 20206 中判断出粒子之间的相似度值大于或等于设置的第二阈值时, 则对所述粒子分别执行变异操作, 以使粒子所对应的任务分配方式中分配给不同处理器的任务相互之间进行交换, 从而避免陷入局部最优解。

- 25 这里, 可以只对所有 $I+1$ 个粒子中两个或两个以上粒子执行变异操作, 而不必对所有 $I+1$ 个粒子执行变异操作。例如, 当有三个粒子且每两个粒子

相互之间的相似度值都大于设置的第二阈值时，可只对所述三个粒子中的任意两个粒子执行变异操作。

步骤 20208：输出粒子群的最优解。

5 作为一种实施方式，将步骤 20204 或步骤 20205 中所述当前的粒子群中的最优粒子所对应的任务分配方式作为所述对染色体种群进行局部寻优后获得的局部最优任务分配方式输出。

步骤 203：判断粒子群的最优解的适应度值是否小于染色体群中最优染色体的适应度值，若是，则继续执行步骤 204，否则执行步骤 205；

10 这里，所述粒子群的最优解即步骤 20208 中所述当前的粒子群中的最优粒子所对应的任务分配方式；所述粒子群的最优解的适应度值为所述最优粒子所对应的任务分配方式下所有 N 个任务在 M 个处理器中完成所需的时间；所述最优染色体为当前的染色体种群中所有 N 个任务在 M 个处理器中完成所需的时间最短的任务分配方式所对应的染色体；所述染色体群中最优染色体的适应度值为所述最优染色体所对应的任务分配方式下所有 N 个任务在 M 个处理器中完成所需的时间。

步骤 204：将所述最优解替换所述染色体群中最优染色体；

作为一种实施方式，将通过粒子群算法所求得的、且具有比所述染色体种群中最优染色体的适应度值小的任务分配方式替换所述最优染色体，以完成对染色体种群的局部寻优更新处理，并继续执行步骤 205。

20 步骤 205：对染色体种群中的染色体执行交叉操作；

这里，所述对染色体种群中的染色体执行交叉操作可以是对染色体种群中全部 Q 条染色体中的两条或两条以上染色体按照遗传算法执行交叉操作，且交叉操作的次数可以是一次或一次以上；当两条染色体执行完一次以上交叉操作后，则将经交叉后生成的所有染色体中所需时间最短的两条染色体作为该两条染色体进行交叉更新后的染色体；当 Q 条染色体中的每两

25

条染色体都执行完一次以上的交叉操作后,将会生成数量大于 Q 的新的染色体,通过计算每一条新的染色体下所有 N 个任务在 M 个处理器中完成所需的时间,将所需时间最短的 Q 条新的染色体作为更新后的染色体种群,以保证更新后获得的染色体种群中包含 Q 条染色体。

- 5 这里,通过对任意两条染色体执行交叉操作,使不同染色体对应的部分任务进行互换即互换分配给处理器的任务,以生成新的染色体,从而完成对染色体种群的更新处理,能够避免陷入早熟收敛。

步骤 206: 判断更新后的染色体种群中最优染色体是否满足要求,若满足,则执行步骤 208, 否则执行步骤 207;

- 10 作为一种实施方式,计算步骤 205 中获得的更新后的染色体种群中每个染色体的适应度值,获得所述更新后的染色体种群中的最优染色体即具有最小适应度值的染色体,判断所述最优染色体的适应度值与所有 N 个任务在 M 个处理器中完成所需的理想时间之差是否小于预设时间阈值,若小于,则执行步骤 208, 否则执行步骤 207。

- 15 这里,所述所有 N 个任务在 M 个处理器中完成所需的理想时间为所有 N 个任务在任意一个处理器中完成所需的时间 T 除以 M ,即所述理想时间可表示为 T/M ; 所述预设时间阈值的大小可以根据实际情况进行设置和调整。

- 20 这里,还可以对步骤 202 至步骤 205 的循环操作次数进行计数,当所述循环操作次数达到一定条件比如大于设置的临界值时,可停止寻优操作并将所述染色体群中最优染色体作为任务调度的最终任务分配方式输出。

步骤 207: 对更新后的染色体种群中染色体执行变异操作,返回步骤 202。

- 25 作为一种实施方式,当步骤 206 中最优染色体的适应度值与所有 N 个任务在 M 个处理器中完成所需的理想时间之差大于或等于预设时间阈值

时，则对步骤 206 中所述更新后的染色体种群中的染色体执行变异操作，然后返回执行步骤 202。

这里，所述对所述更新后的染色体种群中的染色体即任务分配方式执行变异操作的次数可以是对全部 Q 条染色体中的一条或一条以上染色体按照遗传算法执行变异操作，且变异操作的次数可以是一次或一次以上；当一条染色体执行完一次以上变异操作后，则将经变异后生成的所有新的染色体中所需时间最短的染色体作为该染色体进行更新后生成的染色体；当 Q 条染色体中的每一条染色体都执行完一次以上变异操作后，将会生成数量大于 Q 的新的染色体，通过计算每一条染色体下所有 N 个任务在 M 个处理器中完成所需的时间，将所需时间最短的 Q 条染色体作为更新后的染色体种群，以保证更新后获得的染色体种群中包含 Q 条染色体。

这里，当最优染色体即所需时间最短的任务分配方式不满足要求即染色体种群整体陷入局部最优时，通过对任意染色体执行变异操作，从而使染色体种群跳出局部最优继续寻找更优的任务分配方式。

步骤 208: 将所述更新后的染色体种群中最优染色体作为任务调度的最终任务分配方式。

具体地，将步骤 206 中满足要求的所述更新后的染色体种群中最优染色体所对应的任务分配方式作为任务调度的最终任务分配方式，以完成任务调度过程。

为实现上述方法，本发明实施例还提供了一种任务调度装置，图 4 为本发明实施例任务调度装置的组成结构示意图，该装置包括：初始化模块 11、处理模块 12；其中，

所述初始化模块 11，配置为重复执行任务随机分配操作，获得包含有 Q 种任务分配方式的初始处理器分配方案；所述任务随机分配操作为将所有 N 个任务随机分配于 M 个处理器中；

所述处理模块 12，配置为重复执行下述操作，直至找到最终任务分配方式：

基于局部搜索和全局搜索更新处理器分配方案，计算更新后的处理器分配方案中每种任务分配方式下所有 N 个任务在 M 个处理器中完成所需的时间；

当所需时间最短的任务分配方式满足要求时，将所述所需时间最短的任务分配方式作为任务调度的最终任务分配方式。

其中，所述初始化模块 11，具体配置为：重复执行 Q 次任务随机分配操作，获得包含有 Q 种任务分配方式的初始处理器分配方案，即每执行一次任务随机分配操作，则获得一种所有 N 个任务被分配于 M 个处理器中的任务分配方式；所述任务随机分配操作为将所有 N 个任务随机分配于 M 个处理器中。

这里，所述初始化模块 11 将所有 N 个任务随机分配于 M 个处理器中，包括：随机选择 $[1, M]$ 之间的一个整数随机数 m_1 ，并从所有 N 个任务中随机选择一个任务分配给第 m_1 个处理器；随机选择 $[1, M]$ 之间的一个整数随机数 m_2 ，并从剩余的 $N-1$ 个任务中随机选择一个任务分配给第 m_2 个处理器；以此循环，直至所有 N 个任务都被分配给 M 个处理器，以获得一种所有 N 个任务被分配于 M 个处理器中的任务分配方式。

这里，所述 M 、 N 、 Q 为自然数且 $M < N$ ；任务之间都是独立的，每个任务所具有的负载可能相同或不相同，处理器之间也都是独立的，负载的大小影响处理器对任务的完成时间；在实际应用中，为保证任务分配方式之间的差异性而提高泛化性， Q 可设置为大于或等于 M ；由于受到随机因素和 Q 的大小的影响，所述初始处理器分配方案中包含的 Q 种任务分配方式中可能存在相同的任务分配方式，因此，可将相同的任务分配方式从所述初始处理器分配方案中删除，以提高所述初始处理器分配方案中任务分配

方式的多样性。

这里，所述所需时间最短的任务分配方式满足要求是指所述所需时间最短的任务分配方式下所有 N 个任务在 M 个处理器中完成所需的时间与所有 N 个任务在 M 个处理器中完成所需的理想时间之差小于或等于预设时间
5 阈值；所述所有 N 个任务在 M 个处理器中完成所需的理想时间为所有 N 个任务在任意一个处理器中完成所需的时间 T 除以 M ，即所述理想时间可表示为 T/M 。

所述处理模块 12 基于局部搜索和全局搜索更新处理器分配方案，为：

利用粒子群算法对当前的处理器分配方案进行局部搜索，获取所述当
10 前的处理器分配方案中的局部最优任务分配方式；

将所述当前的处理器分配方案中的局部最优任务分配方式下所有 N 个任务在 M 个处理器中完成所需的时间作为第一时间；所述当前的处理器分配方案中的全局最优任务分配方式下所有 N 个任务在 M 个处理器中完成所需的时间作为第二时间；

15 确定所述第一时间小于或等于所述第二时间时，将所述当前的处理器分配方案中的全局最优任务分配方式替换为所述当前的处理器分配方案中的局部最优任务分配方式；

对所述当前的处理器分配方案中的任务分配方式执行交叉操作，获得更新后的处理器分配方案。

20 这里，所述当前的处理器分配方案可能是所述初始化模块 11 生成的初始处理器分配方案，也可能是对所述初始化模块 11 生成的初始处理器分配方案进行一次或一次以上更新后形成的处理器分配方案，但所述当前的处理器分配方案中都包含有 Q 种任务分配方式；此外，对所述当前的处理器分配方案中的任务分配方式执行交叉操作可以是对全部 Q 种任务分配方式中的
25 的两种或两种以上任务分配方式按照遗传算法执行交叉操作，且交叉操作

的次数可以是一次或一次以上；当两种任务分配方式执行完一次以上交叉操作后，则将经交叉后生成的所有新的任务分配方式中所需时间最短的两种任务分配方式作为该两种任务分配方式进行更新后生成的任务分配方式；当 Q 种任务分配方式中的每两种任务分配方式都执行完一次以上的交叉操作后，将会生成数量大于 Q 的新的任务分配方式，通过计算每一种新的任务分配方式下所有 N 个任务在 M 个处理器中完成所需的时间，将所需时间最短的 Q 种新的任务分配方式作为更新后的处理器分配方案，以保证更新后获得的处理器分配方案中包含 Q 种任务分配方式。

这里，通过对任意两种任务分配方式执行交叉操作，使不同任务分配方式中的部分任务进行互换即互换分配给不同处理器的任务，以生成新的任务分配方式，从而完成对处理器分配方案的更新处理，能够避免陷入早熟收敛。

所述处理模块 12 利用粒子群算法对当前的处理器分配方案进行局部搜索，获取所述当前的处理器分配方案中的局部最优任务分配方式，为：

随机生成一个包含有 I 个粒子且每个粒子为一种任务分配方式的初始粒子群，并将所述当前的处理器分配方案中的全局最优任务分配方式作为所述初始粒子群的初始最优粒子加入到所述初始粒子群中；

根据 $V_i(k+1) = V_i(k) \otimes X_{gbest}(k) \otimes X_{ibest}(k)$ 更新在第 $k+1$ 次迭代时粒子 i 的速度 $V_i(k+1)$ ， $1 \leq i \leq I+1$ ， $0 \leq k \leq K-1$ ；其中， $X_{gbest}(k)$ 表示在第 $k+1$ 次迭代之前所述粒子群中最优粒子的位置， $X_{ibest}(k)$ 表示在第 $k+1$ 次迭代之前粒子 i 的最优位置， $\otimes$ 表示交叉操作， K 为设置的最大迭代次数；所述最优粒子为粒子群中具有最小适应度值的粒子；所述适应度值为粒子在所对应的任务分配方式下所有 N 个任务在 M 个处理器中完成所需的时间；

根据 $X_i(k+1) = X_i(k) \otimes V_i(k+1)$ 更新在第 $k+1$ 次迭代时粒子 i 的位置 $X_i(k+1)$ ；

计算粒子 i 在第 $k+1$ 次迭代时的适应度值，并根据粒子 i 在第 $k+1$ 次迭代时的适应度值更新所述粒子 i 在第 $k+1$ 次迭代后的最优位置 $X_{ibest}(k+1)$ 、以及更新在第 $k+1$ 次迭代后所述粒子群中的最优粒子和所述粒子群中更新后的最优粒子的位置 $X_{gbest}(k+1)$ ；

- 5 当检测出所述迭代次数 $k+1$ 小于 K 且所述粒子群中最优粒子的适应度值小于设置的第一阈值或所述迭代次数 $k+1$ 等于 K 时，将所述粒子群中的最优粒子所对应的任务分配方式作为所述当前的处理器分配方案中的局部最优任务分配方式；

- 10 当检测出所述迭代次数 $k+1$ 小于 K 且所述粒子群中最优粒子的适应度值大于或等于设置的第一阈值时，计算第 $k+1$ 次迭代后所述粒子群中粒子之间的相似度值，若所述相似度值大于或等于设置的第二阈值，则对所述粒子的位置执行变异操作。

- 15 这里，除了将所述当前的处理器分配方案中的全局最优任务分配方式加入到所述初始粒子群中以外，还可以将所述当前的处理器分配方案中所需时间较少的多种任务分配方式都加入到所述初始粒子群中，并以所述当前的处理器分配方案中的全局最优任务分配方式作为所述初始粒子群的初始最优粒子；例如，可将所述当前的处理器分配方案中的前 5 种所需时间最短的任务分配方式加入所述初始粒子群中。

- 20 这里，当 $k=0$ 时， $X_{gbest}(0)$ 表示所述初始粒子群中最优粒子的位置，即所述初始粒子群中最优粒子对应的任务分配方式； $X_{ibest}(0)$ 表示所述初始粒子群中粒子 i 的初始最优位置，即所述初始粒子群中粒子 i 对应的任务分配方式； $V_i(0)$ 表示所述初始粒子群中粒子 i 的初始速度，为 $(T_i(0)*M*v)/T_i(0)$ ；其中， $T_i(0)$ 表示所述初始粒子群中粒子 i 对应的任务分配方式下所有 N 个任务在 M 个处理器中完成所需的时间， $T_1(0)$ 表示所述初始粒子群中最优粒子对应的任务分配方式下所有 N 个任务在 M 个处理器中完成所需的时间， v 表
- 25

示速度参数。

这里，当对所述当前的粒子群中的粒子的速度和位置进行更新后，粒子对应的任务分配方式中给每个处理器分配的任务数量发生变化，所述当前的粒子群中的粒子都向所述当前的粒子群中的最优粒子靠近，即粒子对应的任务分配方式都朝向最优粒子对应的任务分配方式靠近，以使粒子对应的任务分配方式中每个处理器的任务负载趋向于相等。

这里，当第 $k+1$ 次迭代后所生成的粒子群中粒子 i 的适应度值小于第 k 次迭代后所生成的粒子群中粒子 i 的适应度值时，则保持粒子 i 在第 $k+1$ 次迭代后的最优位置 $X_{ibest}(k+1)$ 为粒子 i 在第 k 次迭代后的最优位置 $X_{ibest}(k)$ ；当第 $k+1$ 次迭代后所生成的粒子群中最优粒子的适应度值小于第 k 次迭代后所生成的粒子群中最优粒子的适应度值时，则第 $k+1$ 次迭代后所生成的粒子群中最优粒子仍然保持为第 k 次迭代后所生成的粒子群中最优粒子、第 $k+1$ 次迭代后所生成的粒子群中最优粒子的位置 $X_{gbest}(k+1)$ 仍然保持为第 k 次迭代后所生成的粒子群中最优粒子的位置 $X_{gbest}(k)$ 。

这里，根据公式 $similarity(X_{ij}) = \sum_{f=1}^M same(X_i^f, X_j^f)$ ， $same(x, y) = \begin{cases} 0, & \text{if } x = y \\ 1, & \text{if } x \neq y \end{cases}$ 计算所述粒子的位置相互之间的相似度值 $similarity(X_{ij})$ ；其中， X_i^f 、 X_j^f 分别代表粒子 i 和粒子 j 对应的任务分配方式中相同处理器被分配的任务。

这里，可以计算所有 $I+1$ 个粒子中两个或两个以上粒子之间的相似度值，比如，当计算出两个粒子之间的相似度值大于设置的第二阈值时，可认为是这两个粒子分别所对应的任务分配方式中将相同的任务分配给了相同的处理器；若两个粒子之间的相似度值大于设置的第二阈值时，可对这两个粒子分别执行变异操作，以使粒子所对应的任务分配方式中分配给不同处理器的任务进行互换，从而避免陷入局部最优解。

这里，可以只对所有 $I+1$ 个粒子中两个或两个以上粒子执行变异操作，

而不必对所有 $I+1$ 个粒子执行变异操作。例如，当有三个粒子且每两个粒子相互之间的相似度值都大于设置的第二阈值时，可只对所述三个粒子中的任意两个粒子执行变异操作。

所述处理模块 12 随机生成一个包含有 I 个粒子且每个粒子为一种任务分配方式的初始粒子群，包括：

从所有 N 个任务中随机选择一个任务分配给第一个处理器；从剩余的 $N-1$ 个任务中随机选择一个任务分配给所述第二个处理器；从剩余的 $N-m$ 个任务中随机选择一个任务分配给所述第 m 个处理器， $3 \leq m \leq M$ ；以此循环，直至所有 N 个任务都被分配给 M 个处理器，获得一个对应一种任务分配方式的粒子；其中，当 m 大于 M 即随机分配了 M 个任务后，重新自第一个处理器开始分配；

循环执行上述分配过程，直至获取到与 I 种任务分配方式一一对应的 I 个粒子。

这里，当 m 大于 M 时，即已将 m 个任务随机分配给 M 个处理器，而每个处理器被分配了一个任务后，重新自第一个处理器开始分配任务，即从剩余的 $N-M$ 个任务中随机选择一个任务分配给所述第一个处理器，从剩余的 $N-M-1$ 个任务中随机选择一个任务分配给所述第二个处理器，并以此类推。

作为一种实施方式，所述处理模块 12，还配置为当所需时间最短的任务分配方式不满足要求时，对所述更新后的处理器分配方案中的任务分配方式执行变异操作。

作为一种实施方式，当获取的所需时间最短的任务分配方式不满足要求时，即所述所需时间最短的任务分配方式下所有 N 个任务在 M 个处理器中完成所需的时间与所有 N 个任务在 M 个处理器中完成所需的理想时间之差大于预设时间阈值，则对所述更新后的处理器分配方案中的任务分配方

式执行变异操作，然后继续利用粒子群算法进行局部寻优。

这里，所述对所述更新后的处理器分配方案中的任务分配方式执行变异操作的次数可以是对全部 Q 种任务分配方式中的一种或一种以上任务分配方式按照遗传算法执行变异操作，且变异操作的次数可以是一次或一次以上；当一种任务分配方式执行完一次以上变异操作后，则将经变异后生成的所有新的任务分配方式中所需时间最短的任务分配方式作为该任务分配方式进行更新后的任务分配方式；当 Q 种任务分配方式中的每一种任务分配方式都执行完一次以上变异操作后，将会生成数量大于 Q 的新的任务分配方式，通过计算每一个新的任务分配方式下所有 N 个任务在 M 个处理器中完成所需的时间，将所需时间最短的 Q 种新的任务分配方式作为更新后的任务分配方式，以保证更新后获得的处理器分配方案中包含 Q 种任务分配方式。

这里，当所需时间最短的任务分配方式不满足要求即粒子群整体陷入局部最优时，通过对任意任务分配方式执行变异操作，改变粒子的位置和速度，从而使粒子群整体跳出局部最优继续寻优。

在实际应用中，所述初始化模块 11、处理模块 12 均可由位于终端中的中央处理器（CPU, Central Processing Unit）、微处理器（MPU, Microprocessor Unit）、数字信号处理器（DSP, Digital Signal Processor）、或现场可编程门阵列（FPGA, Field - Programmable Gate Array）等实现。

本实施例还提供一种存储介质，所述存储介质存储有计算机程序，所述计算机程序被处理器执行之后，能够实现前述任意一项或多项任务调度方法。

所述计算机存储介质可为各种类型的存储介质，在本实施例中可优选为非瞬间存储介质。

本领域技术人员应当理解，本实施例的存储介质中各程序的功能，可

参照实施例所述的任务调度方法的相关描述而理解。

本领域内的技术人员应明白，本发明的实施例可提供为方法、系统、或计算机程序产品。因此，本发明可采用硬件实施例、软件实施例、或结合软件和硬件方面的实施例的形式。而且，本发明可采用在一个或多个其中包含有计算机可用程序代码的计算机可用存储介质（包括但不限于磁盘存储器、5 光学存储器等）上实施的计算机程序产品的形式。

本发明是参照根据本发明实施例的方法、设备（系统）、和计算机程序产品的流程图和/或方框图来描述的。应理解可由计算机程序指令实现流程图和/或方框图中的每一流程和/或方框、以及流程图和/或方框图中的流程和/或方框的结合。可提供这些计算机程序指令到通用计算机、专用计算机、10 嵌入式处理机或其他可编程数据处理设备的处理器以产生一个机器，使得通过计算机或其他可编程数据处理设备的处理器执行的指令产生用于实现在流程图一个流程或多个流程和/或方框图一个方框或多个方框中指定的功能的装置。

15 这些计算机程序指令也可存储在能引导计算机或其他可编程数据处理设备以特定方式工作的计算机可读存储器中，使得存储在该计算机可读存储器中的指令产生包括指令装置的制造品，该指令装置实现在流程图一个流程或多个流程和/或方框图一个方框或多个方框中指定的功能。

20 这些计算机程序指令也可装载到计算机或其他可编程数据处理设备上，使得在计算机或其他可编程设备上执行一系列操作步骤以产生计算机实现的处理，从而在计算机或其他可编程设备上执行的指令提供用于实现在流程图一个流程或多个流程和/或方框图一个方框或多个方框中指定的功能的步骤。

25 以上所述，仅为本发明的较佳实施例而已，并非用于限定本发明的保护范围。凡在本发明的精神和范围之内所作的任何修改、等同替换和改进

等，均包含在本发明的保护范围之内。

工业实用性

本发明实施例的技术方案通过将在全局空间搜索全局最优解和在局部空间搜索局部最优解结合起来，两者相互协调和促进，以求取最优的任务分配方式。本发明实施例的技术方案利用遗传算法求解任务调度的全局最优任务分配方式，而利用粒子群算法求解任务调度的局部最优任务分配方式，并将局部最优任务分配方式作为全局最优任务分配方式进化的依据，能够在粒子群整体陷入局部最优时，通过变异操作改变粒子的当前位置和速度，解决了粒子群算法在求解调度问题时容易出现早熟收敛、易陷入局部最优等缺陷。

权利要求书

1、一种任务调度方法，所述方法包括：

重复执行任务随机分配操作，获得包含有 Q 种任务分配方式的初始处理器分配方案；所述任务随机分配操作为将所有 N 个任务随机分配于 M 个处理器中； M 、 N 、 Q 为自然数且 $M < N$ ；

重复执行下述操作，直至找到最终任务分配方式：

基于局部搜索和全局搜索更新处理器分配方案，计算更新后的处理器分配方案中每种任务分配方式下所有 N 个任务在 M 个处理器中完成所需的时间；

当所需时间最短的任务分配方式满足要求时，将所述所需时间最短的任务分配方式作为任务调度的最终任务分配方式。

2、根据权利要求 1 所述的方法，其中，所述方法还包括：

当所需时间最短的任务分配方式不满足要求时，对所述更新后的处理器分配方案中的任务分配方式执行变异操作。

3、根据权利要求 1 或 2 所述的方法，其中，所述将所有 N 个任务随机分配于 M 个处理器中，包括：

随机选择 $[1, M]$ 之间的一个整数随机数 m_1 ，并从所有 N 个任务中随机选择一个任务分配给第 m_1 个处理器；随机选择 $[1, M]$ 之间的一个整数随机数 m_2 ，并从剩余的 $N-1$ 个任务中随机选择一个任务分配给第 m_2 个处理器；以此循环，直至所有 N 个任务都被分配给 M 个处理器，获得一种任务分配方式。

4、根据权利要求 1 或 2 所述的方法，其中，所述基于局部搜索和全局搜索更新处理器分配方案，包括：

利用粒子群算法对当前的处理器分配方案进行局部搜索，获取所述当

前的处理器分配方案中的局部最优任务分配方式;

将所述当前的处理器分配方案中的局部最优任务分配方式下所有 N 个任务在 M 个处理器中完成所需的时间作为第一时间; 所述当前的处理器分配方案中的全局最优任务分配方式下所有 N 个任务在 M 个处理器中完成所需的时间作为第二时间;

确定所述第一时间小于或等于所述第二时间时, 将所述当前的处理器分配方案中的全局最优任务分配方式替换为所述当前的处理器分配方案中的局部最优任务分配方式;

对所述当前的处理器分配方案中的任务分配方式执行交叉操作, 获得更新后的处理器分配方案。

5、根据权利要求 4 所述的方法, 其中, 所述利用粒子群算法对当前的处理器分配方案进行局部搜索, 获取所述当前的处理器分配方案中的局部最优任务分配方式, 包括:

随机生成一个包含有 I 个粒子且每个粒子为一种任务分配方式的初始粒子群, 并将所述当前的处理器分配方案中的全局最优任务分配方式作为所述初始粒子群的初始最优粒子加入到所述初始粒子群中;

根据 $V_i(k+1) = V_i(k) \otimes X_{gbest}(k) \otimes X_{ibest}(k)$ 更新在第 $k+1$ 次迭代时粒子 i 的速度 $V_i(k+1)$, $1 \leq i \leq I+1$, $0 \leq k \leq K-1$; 其中, $X_{gbest}(k)$ 表示在第 $k+1$ 次迭代之前所述粒子群中最优粒子的位置, $X_{ibest}(k)$ 表示在第 $k+1$ 次迭代之前粒子 i 的最优位置, $\otimes$ 表示交叉操作, K 为设置的最大迭代次数; 所述最优粒子为粒子群中具有最小适应度值的粒子; 所述适应度值为粒子在所对应的任务分配方式下所有 N 个任务在 M 个处理器中完成所需的时间;

根据 $X_i(k+1) = X_i(k) \otimes V_i(k+1)$ 更新在第 $k+1$ 次迭代时粒子 i 的位置 $X_i(k+1)$;

计算粒子 i 在第 $k+1$ 次迭代时的适应度值, 并根据粒子 i 在第 $k+1$ 次迭代

时的适应度值更新所述粒子 i 在第 $k+1$ 次迭代后的最优位置 $X_{ibest}(k+1)$ 、以及更新在第 $k+1$ 次迭代后所述粒子群中的最优粒子和所述粒子群中更新后的最优粒子的位置 $X_{gbest}(k+1)$;

5 当检测出所述迭代次数 $k+1$ 小于 K 且所述粒子群中最优粒子的适应度值小于设置的第一阈值或所述迭代次数 $k+1$ 等于 K 时,将所述粒子群中的最优粒子所对应的任务分配方式作为所述当前的处理器分配方案中的局部最优任务分配方式;

10 当检测出所述迭代次数 $k+1$ 小于 K 且所述粒子群中最优粒子的适应度值大于或等于设置的第一阈值时,计算第 $k+1$ 次迭代后所述粒子群中粒子之间的相似度值,若所述相似度值大于或等于设置的第二阈值,则对所述粒子的位置执行变异操作。

6、根据权利要求5所述的方法,其中,所述随机生成一个包含有 I 个粒子且每个粒子为一种任务分配方式的初始粒子群,包括:

15 从所有 N 个任务中随机选择一个任务分配给第一个处理器;从剩余的 $N-1$ 个任务中随机选择一个任务分配给所述第二个处理器;从剩余的 $N-m$ 个任务中随机选择一个任务分配给所述第 m 个处理器, $3 \leq m \leq M$;以此循环,直至所有 N 个任务都被分配给 M 个处理器,获得一个对应一种任务分配方式的粒子;其中,当 m 大于 M 即随机分配了 M 个任务后,重新自第一个处理器开始分配;

20 循环执行上述分配过程,直至获取到与 I 种任务分配方式一一对应的 I 个粒子。

7、一种任务调度装置,所述装置包括:初始化模块、处理模块;其中,

所述初始化模块,配置为重复执行任务随机分配操作,获得包含有 Q 种任务分配方式的初始处理器分配方案;所述任务随机分配操作为将所有 N
25 个任务随机分配于 M 个处理器中; M 、 N 、 Q 为自然数且 $M < N$;

所述处理模块，配置为重复执行下述操作，直至找到最终任务分配方式：

基于局部搜索和全局搜索更新处理器分配方案，计算更新后的处理器分配方案中每种任务分配方式下所有 N 个任务在 M 个处理器中完成所需的时间；

当所需时间最短的任务分配方式满足要求时，将所述所需时间最短的任务分配方式作为任务调度的最终任务分配方式。

8、根据权利要求 7 所述的装置，其中，所述处理模块，还配置为当所需时间最短的任务分配方式不满足要求时，对所述更新后的处理器分配方案中的任务分配方式执行变异操作。

9、根据权利要求 7 或 8 所述的装置，其中，所述初始化模块将所有 N 个任务随机分配于 M 个处理器中为：

随机选择 $[1, M]$ 之间的一个整数随机数 m_1 ，并从所有 N 个任务中随机选择一个任务分配给第 m_1 个处理器；随机选择 $[1, M]$ 之间的一个整数随机数 m_2 ，并从剩余的 $N-1$ 个任务中随机选择一个任务分配给第 m_2 个处理器；以此循环，直至所有 N 个任务都被分配给 M 个处理器，获得一种任务分配方式。

10、根据权利要求 7 或 8 所述的装置，其中，所述处理模块基于局部搜索和全局搜索更新处理器分配方案为：

利用粒子群算法对当前的处理器分配方案进行局部搜索，获取所述当前的处理器分配方案中的局部最优任务分配方式；

将所述当前的处理器分配方案中的局部最优任务分配方式下所有 N 个任务在 M 个处理器中完成所需的时间作为第一时间；所述当前的处理器分配方案中的全局最优任务分配方式下所有 N 个任务在 M 个处理器中完成所需的时间作为第二时间；

确定所述第一时间小于或等于所述第二时间时, 将所述当前的处理器分配方案中的全局最优任务分配方式替换为所述当前的处理器分配方案中的局部最优任务分配方式;

对所述当前的处理器分配方案中的任务分配方式执行交叉操作, 获得更新后的处理器分配方案。

11、根据权利要求 10 所述的装置, 其中, 所述处理模块利用粒子群算法对当前的处理器分配方案进行局部搜索, 获取所述当前的处理器分配方案中的局部最优任务分配方式为:

随机生成一个包含有 I 个粒子且每个粒子为一种任务分配方式的初始粒子群, 并将所述当前的处理器分配方案中的全局最优任务分配方式作为所述初始粒子群的初始最优粒子加入到所述初始粒子群中;

根据 $V_i(k+1) = V_i(k) \otimes X_{gbest}(k) \otimes X_{ibest}(k)$ 更新在第 $k+1$ 次迭代时粒子 i 的速度 $V_i(k+1)$, $1 \leq i \leq I+1$, $0 \leq k \leq K-1$; 其中, $X_{gbest}(k)$ 表示在第 $k+1$ 次迭代之前所述粒子群中最优粒子的位置, $X_{ibest}(k)$ 表示在第 $k+1$ 次迭代之前粒子 i 的最优位置, $\otimes$ 表示交叉操作, K 为设置的最大迭代次数; 所述最优粒子为粒子群中具有最小适应度值的粒子; 所述适应度值为粒子在所对应的任务分配方式下所有 N 个任务在 M 个处理器中完成所需的时间;

根据 $X_i(k+1) = X_i(k) \otimes V_i(k+1)$ 更新在第 $k+1$ 次迭代时粒子 i 的位置 $X_i(k+1)$;

计算粒子 i 在第 $k+1$ 次迭代时的适应度值, 并根据粒子 i 在第 $k+1$ 次迭代时的适应度值更新所述粒子 i 在第 $k+1$ 次迭代后的最优位置 $X_{ibest}(k+1)$ 、以及更新在第 $k+1$ 次迭代后所述粒子群中的最优粒子和所述粒子群中更新后的最优粒子的位置 $X_{gbest}(k+1)$;

当检测出所述迭代次数 $k+1$ 小于 K 且所述粒子群中最优粒子的适应度值小于设置的第一阈值或所述迭代次数 $k+1$ 等于 K 时, 将所述粒子群中的最

优粒子所对应的任务分配方式作为所述当前的处理器分配方案中的局部最优任务分配方式;

当检测出所述迭代次数 $k+1$ 小于 K 且所述粒子群中最优粒子的适应度值大于或等于设置的第一阈值时,计算第 $k+1$ 次迭代后所述粒子群中粒子之间的相似度值,若所述相似度值大于或等于设置的第二阈值,则对所述粒子的位置执行变异操作。

12、根据权利要求 11 所述的装置,其中,所述处理模块随机生成一个包含有 I 个粒子且每个粒子为一种任务分配方式的初始粒子群为:

从所有 N 个任务中随机选择一个任务分配给第一个处理器;从剩余的 $N-1$ 个任务中随机选择一个任务分配给所述第二个处理器;从剩余的 $N-m$ 个任务中随机选择一个任务分配给所述第 m 个处理器, $3 \leq m \leq M$;以此循环,直至所有 N 个任务都被分配给 M 个处理器,获得一个对应一种任务分配方式的粒子;其中,当 m 大于 M 即随机分配了 M 个任务后,重新自第一个处理器开始分配;

循环执行上述分配过程,直至获取到与 I 种任务分配方式一一对应的 I 个粒子。

13、一种计算机存储介质,所述计算机存储介质中存储有计算机可执行指令,所述计算机可执行指令用于执行权利要求 1 至 6 任一项所述的任务调度方法。

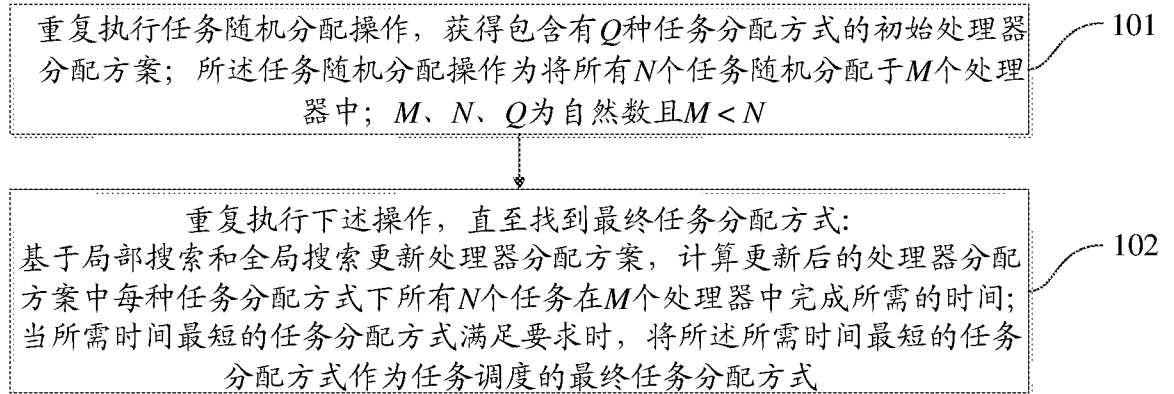


图 1

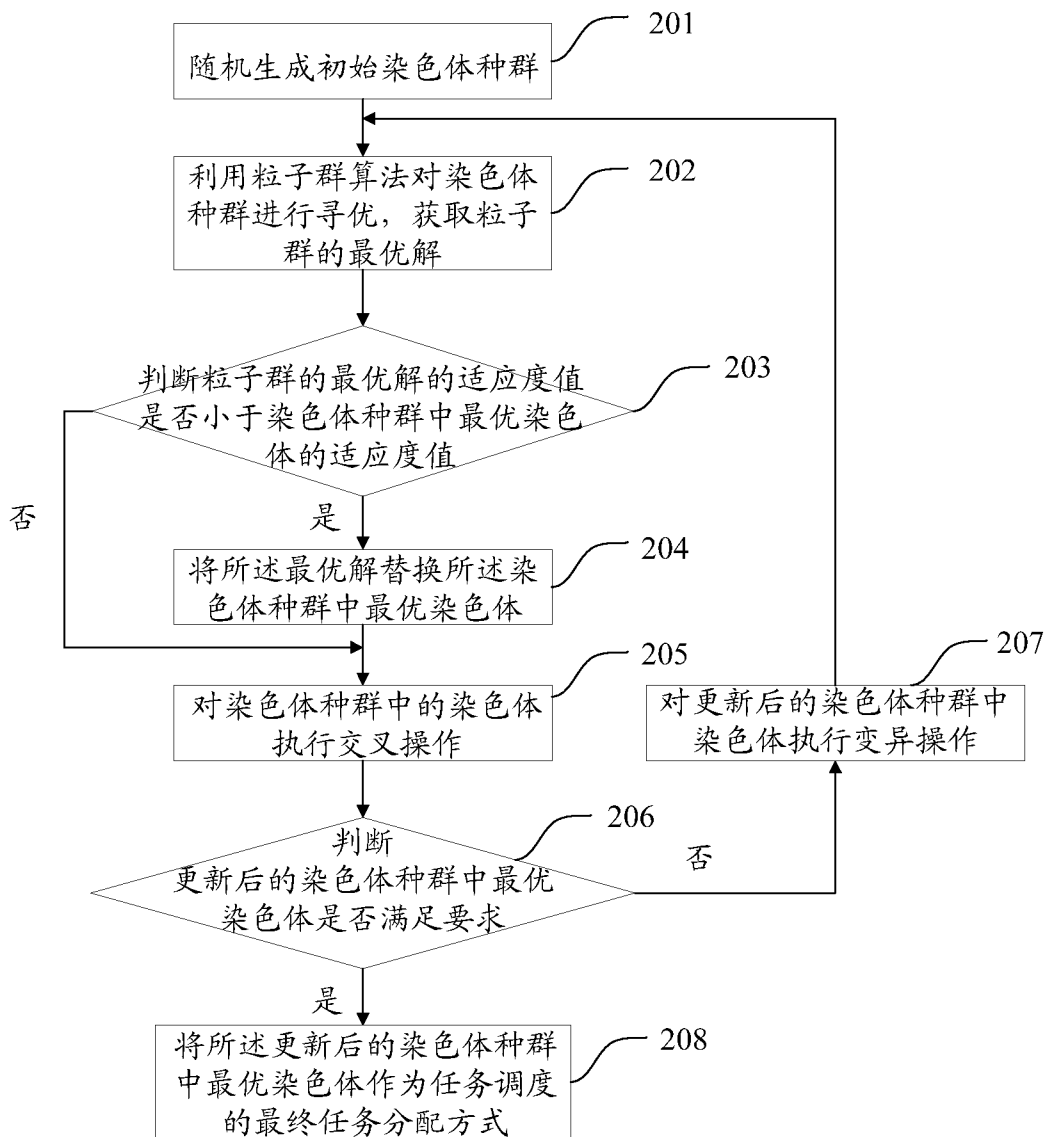


图 2

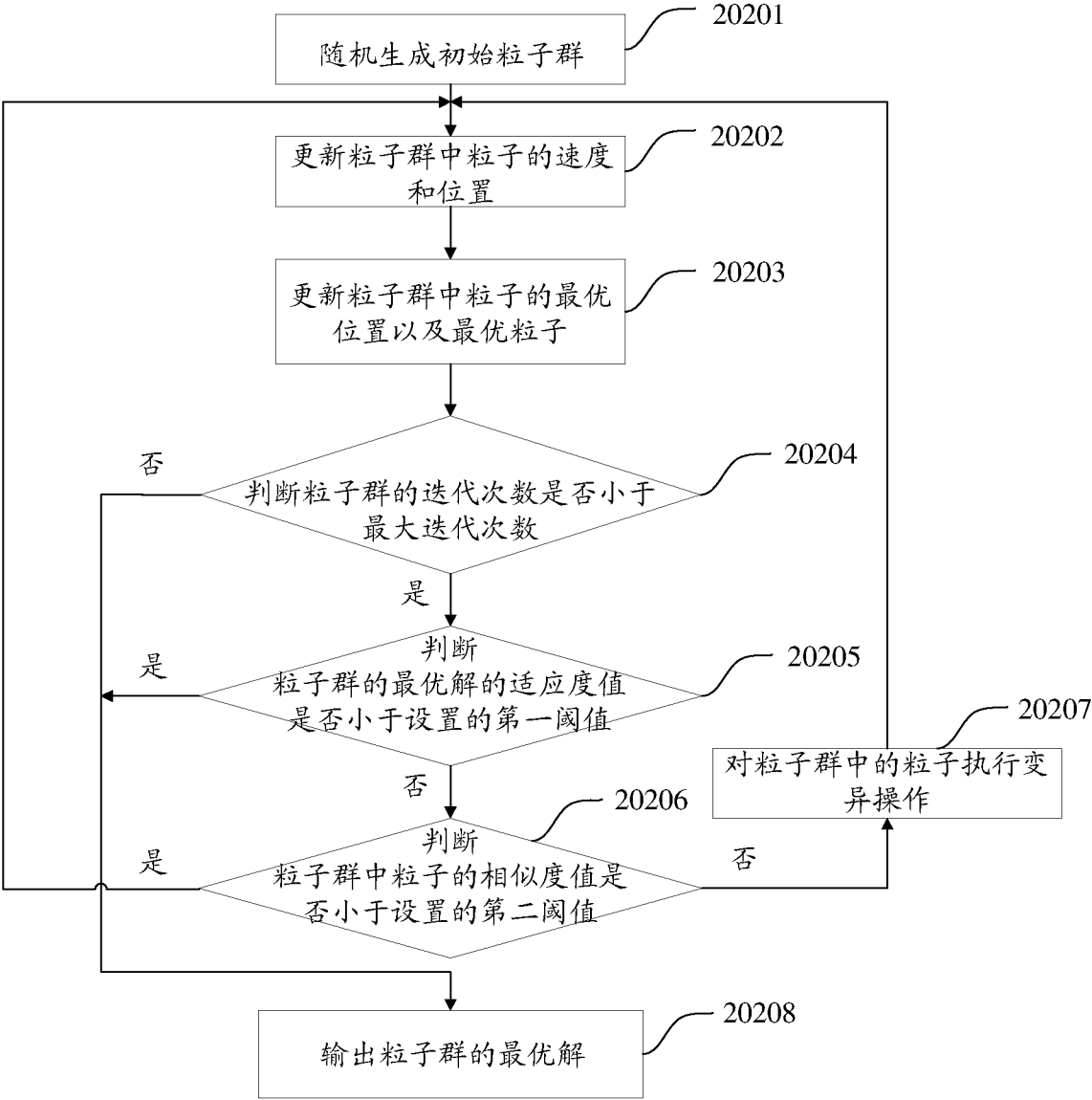


图 3

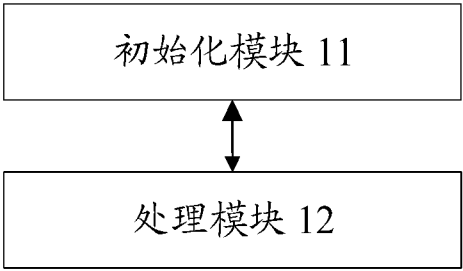


图 4

INTERNATIONAL SEARCH REPORT

International application No.
PCT/CN2017/091802

A. CLASSIFICATION OF SUBJECT MATTER

G06F 9/50 (2006.01) i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

CNPAT, EPODOC, WPI, CNKI: 任务调度, 全局, 局部, 交叉, 变异, 遗传算法, 染色体, 粒子群, task, scheduling, global, local, chiasmatsby, variation, genetic, particle group

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	CN 102662743 A (NANJING UNIVERSITY OF INFORMATION SCIENCE & TECHNOLOGY), 12 September 2012 (12.09.2012), description, paragraphs [0041]-[0097], and figures 1-2	1-4, 7-10, 13
X	CN 102866912 A (CAPITAL NORMAL UNIVERSITY), 09 January 2013 (09.01.2013), description, paragraphs [0084]-[0107], and figure 7	1-4, 7-10, 13
A	CN 101604258 A (HANGZHOU DIANZI UNIVERSITY), 16 December 2009 (16.12.2009), entire document	1-13
A	CN 103699446 A (NANJING UNIVERSITY OF INFORMATION SCIENCE & TECHNOLOGY), 02 April 2014 (02.04.2014), entire document	1-13
A	CN 104765640 A (INSTITUTE OF ACOUSTICS, CHINESE ACADEMY OF SCIENCES), 08 July 2015 (08.07.2015), entire document	1-13

☐ Further documents are listed in the continuation of Box C.

☒ See patent family annex.

* Special categories of cited documents:	“T” later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
“A” document defining the general state of the art which is not considered to be of particular relevance	“X” document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
“E” earlier application or patent but published on or after the international filing date	“Y” document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
“L” document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	“&” document member of the same patent family
“O” document referring to an oral disclosure, use, exhibition or other means	
“P” document published prior to the international filing date but later than the priority date claimed	

Date of the actual completion of the international search 22 September 2017	Date of mailing of the international search report 19 October 2017
Name and mailing address of the ISA State Intellectual Property Office of the P. R. China No. 6, Xitucheng Road, Jimenqiao Haidian District, Beijing 100088, China Facsimile No. (86-10) 62019451	Authorized officer SUN, Juan Telephone No. (86-10) 62413714

International application No.
PCT/CN2017/091802

Form PCT/ISA/210 (patent family annex) (July 2009)

国际检索报告

国际申请号

PCT/CN2017/091802

A. 主题的分类 G06F 9/50 (2006.01) i 按照国际专利分类 (IPC) 或者同时按照国家分类和 IPC 两种分类																				
B. 检索领域 检索的最低限度文献 (标明分类系统和分类号) G06F 包含在检索领域中的除最低限度文献以外的检索文献 在国际检索时查阅的电子数据库 (数据库的名称, 和使用的检索词 (如使用)) CNPAT, EPDOC, WPI, CNKI: 任务调度, 全局, 局部, 交叉, 变异, 遗传算法, 染色体, 粒子群, task, scheduling, global, local, chiasmotypy, variation, genetic, particle group																				
C. 相关文件 <table border="1"> <thead> <tr> <th>类 型*</th> <th>引用文件, 必要时, 指明相关段落</th> <th>相关的权利要求</th> </tr> </thead> <tbody> <tr> <td>X</td> <td>CN 102662743 A (南京信息工程大学) 2012年 9月 12日 (2012 - 09 - 12) 说明书第[0041]-[0097]段, 附图1-2</td> <td>1-4, 7-10, 13</td> </tr> <tr> <td>X</td> <td>CN 102866912 A (首都师范大学) 2013年 1月 9日 (2013 - 01 - 09) 说明书第[0084]-[0107]段, 附图7</td> <td>1-4, 7-10, 13</td> </tr> <tr> <td>A</td> <td>CN 101604258 A (杭州电子科技大学) 2009年 12月 16日 (2009 - 12 - 16) 全文</td> <td>1-13</td> </tr> <tr> <td>A</td> <td>CN 103699446 A (南京信息工程大学) 2014年 4月 2日 (2014 - 04 - 02) 全文</td> <td>1-13</td> </tr> <tr> <td>A</td> <td>CN 104765640 A (中国科学院声学研究所) 2015年 7月 8日 (2015 - 07 - 08) 全文</td> <td>1-13</td> </tr> </tbody> </table>			类 型*	引用文件, 必要时, 指明相关段落	相关的权利要求	X	CN 102662743 A (南京信息工程大学) 2012年 9月 12日 (2012 - 09 - 12) 说明书第[0041]-[0097]段, 附图1-2	1-4, 7-10, 13	X	CN 102866912 A (首都师范大学) 2013年 1月 9日 (2013 - 01 - 09) 说明书第[0084]-[0107]段, 附图7	1-4, 7-10, 13	A	CN 101604258 A (杭州电子科技大学) 2009年 12月 16日 (2009 - 12 - 16) 全文	1-13	A	CN 103699446 A (南京信息工程大学) 2014年 4月 2日 (2014 - 04 - 02) 全文	1-13	A	CN 104765640 A (中国科学院声学研究所) 2015年 7月 8日 (2015 - 07 - 08) 全文	1-13
类 型*	引用文件, 必要时, 指明相关段落	相关的权利要求																		
X	CN 102662743 A (南京信息工程大学) 2012年 9月 12日 (2012 - 09 - 12) 说明书第[0041]-[0097]段, 附图1-2	1-4, 7-10, 13																		
X	CN 102866912 A (首都师范大学) 2013年 1月 9日 (2013 - 01 - 09) 说明书第[0084]-[0107]段, 附图7	1-4, 7-10, 13																		
A	CN 101604258 A (杭州电子科技大学) 2009年 12月 16日 (2009 - 12 - 16) 全文	1-13																		
A	CN 103699446 A (南京信息工程大学) 2014年 4月 2日 (2014 - 04 - 02) 全文	1-13																		
A	CN 104765640 A (中国科学院声学研究所) 2015年 7月 8日 (2015 - 07 - 08) 全文	1-13																		
☐ 其余文件在C栏的续页中列出。 ☒ 见同族专利附件。																				
<table border="0"> <tr> <td> * 引用文件的具体类型: “A” 认为不特别相关的表示了现有技术一般状态的文件 “E” 在国际申请日的当天或之后公布的在先申请或专利 “L” 可能对优先权要求构成怀疑的文件, 或为确定另一篇引用文件的公布日而引用的或者因其他特殊理由而引用的文件 (如具体说明的) “O” 涉及口头公开、使用、展览或其他方式公开的文件 “P” 公布日先于国际申请日但迟于所要求的优先权日的文件 </td> <td> “T” 在申请日或优先权日之后公布, 与申请不相抵触, 但为了理解发明之理论或原理的在后文件 “X” 特别相关的文件, 单独考虑该文件, 认定要求保护的发明不是新颖的或不具有创造性 “Y” 特别相关的文件, 当该文件与另一篇或者多篇该类文件结合并且这种结合对于本领域技术人员为显而易见时, 要求保护的发明不具有创造性 “&” 同族专利的文件 </td> </tr> </table>			* 引用文件的具体类型: “A” 认为不特别相关的表示了现有技术一般状态的文件 “E” 在国际申请日的当天或之后公布的在先申请或专利 “L” 可能对优先权要求构成怀疑的文件, 或为确定另一篇引用文件的公布日而引用的或者因其他特殊理由而引用的文件 (如具体说明的) “O” 涉及口头公开、使用、展览或其他方式公开的文件 “P” 公布日先于国际申请日但迟于所要求的优先权日的文件	“T” 在申请日或优先权日之后公布, 与申请不相抵触, 但为了理解发明之理论或原理的在后文件 “X” 特别相关的文件, 单独考虑该文件, 认定要求保护的发明不是新颖的或不具有创造性 “Y” 特别相关的文件, 当该文件与另一篇或者多篇该类文件结合并且这种结合对于本领域技术人员为显而易见时, 要求保护的发明不具有创造性 “&” 同族专利的文件																
* 引用文件的具体类型: “A” 认为不特别相关的表示了现有技术一般状态的文件 “E” 在国际申请日的当天或之后公布的在先申请或专利 “L” 可能对优先权要求构成怀疑的文件, 或为确定另一篇引用文件的公布日而引用的或者因其他特殊理由而引用的文件 (如具体说明的) “O” 涉及口头公开、使用、展览或其他方式公开的文件 “P” 公布日先于国际申请日但迟于所要求的优先权日的文件	“T” 在申请日或优先权日之后公布, 与申请不相抵触, 但为了理解发明之理论或原理的在后文件 “X” 特别相关的文件, 单独考虑该文件, 认定要求保护的发明不是新颖的或不具有创造性 “Y” 特别相关的文件, 当该文件与另一篇或者多篇该类文件结合并且这种结合对于本领域技术人员为显而易见时, 要求保护的发明不具有创造性 “&” 同族专利的文件																			
国际检索实际完成的日期	国际检索报告邮寄日期																			
2017年 9月 22日	2017年 10月 19日																			
ISA/CN的名称和邮寄地址	受权官员																			
中华人民共和国国家知识产权局 (ISA/CN) 中国北京市海淀区蓟门桥西土城路6号 100088 传真号 (86-10) 62019451	孙娟 电话号码 (86-10) 62413714																			

国际检索报告
关于同族专利的信息

国际申请号

PCT/CN2017/091802

检索报告引用的专利文件			公布日 (年/月/日)	同族专利	公布日 (年/月/日)
CN	102662743	A	2012年 9月 12日	无	
CN	102866912	A	2013年 1月 9日	无	
CN	101604258	A	2009年 12月 16日	无	
CN	103699446	A	2014年 4月 2日	无	
CN	104765640	A	2015年 7月 8日	WO 2015100995 A1	2015年 7月 9日