



(12) 发明专利申请

(10) 申请公布号 CN 114579110 A

(43) 申请公布日 2022.06.03

(21) 申请号 202210480911.3

(22) 申请日 2022.05.05

(71) 申请人 支付宝(杭州)信息技术有限公司
地址 310000 浙江省杭州市西湖区西溪路
556号8层B段801-11

(72) 发明人 简道红 吴华

(74) 专利代理机构 北京博思佳知识产权代理有
限公司 11415
专利代理师 李威

(51) Int.Cl.
G06F 8/30 (2018.01)

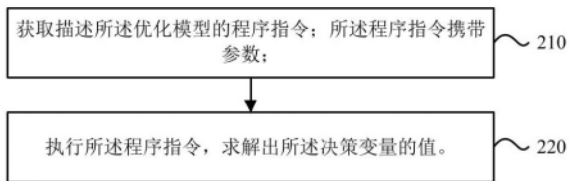
权利要求书2页 说明书9页 附图5页

(54) 发明名称

优化模型的求解方法、电子设备、应用程序及存储介质

(57) 摘要

本说明书提供优化模型的求解方法、电子设备、应用程序及存储介质,优化模型的求解过程包括获取描述优化模型的程序指令,其中,程序指令中携带有参数。在获取程序指令的过程中,用户可以输入描述决策变量的信息,描述优化目标的信息以及描述约束条件的信息。其中,输入信息携带有参数的存储位置信息。如此,可以基于存储位置信息读取到程序指令中的参数,并且根据预设模板记录的程序指令转换方式,将输入信息转换为描述优化模型的程序指令。由于用户只需输入上述三种信息,不需要使用建模语言将优化模型编写成计算机程序,因此用户可以不用掌握建模语言,降低了模型建立门槛。同时节省了用户编写计算机程序的时间,因此提高了业务问题解决效率。



1. 一种优化模型的求解方法,所述优化模型包括决策变量、优化目标和约束条件;所述方法包括:

获取描述所述优化模型的程序指令;所述程序指令包括:用于描述所述决策变量的程序指令、用于描述所述优化目标的程序指令、以及用于描述所述约束条件的程序指令;所述程序指令携带参数;

执行所述程序指令,求解出所述决策变量的值;

其中,所述程序指令的获取步骤包括:

获取用户的输入信息,所述输入信息包括:用于描述所述决策变量的信息、用于描述所述优化目标的信息、以及用于描述所述约束条件的信息;所述输入信息携带所述参数的存储位置信息;

基于所述存储位置信息,读取所述参数;

根据预设模板记录的程序指令转换方式以及所读取的参数,将所述输入信息转换为描述所述优化模型的程序指令。

2. 根据权利要求1所述的方法,所述参数存储在数据源中,所述输入信息携带所述参数的存储位置信息;包括:

所述输入信息携带数据源标识和所述存储位置信息;所述数据源标识用于表示参数从数据源中确定。

3. 根据权利要求1所述的方法,

所述用于描述所述优化目标的信息,包括优化目标表达式;所述优化目标表达式包括预定义的优化目标字符串;

所述用于描述所述约束条件的信息,包括约束条件表达式;所述约束条件表达式包括预定义的约束条件字符串。

4. 根据权利要求1-3任一所述的方法,所述用于描述所述决策变量的信息,包括:

决策变量表达式;所述决策变量表达式包括预定义的决策变量字符串;和/或

赋值所述决策变量的参数的存储位置信息。

5. 根据权利要求4所述的方法,在所述用于描述所述决策变量的信息包括赋值所述决策变量的参数的存储位置信息的情况下,所述执行所述程序指令,求解出所述决策变量的值,包括:

在执行所述程序指令时,基于赋值所述决策变量的参数的存储位置,读取赋值所述决策变量的参数;

基于读取的参数,求解所述决策变量的值。

6. 根据权利要求1所述的方法,所述输入信息携带运算符;所述预设模板还记录有所述运算符与运算公式的转换关系,所述根据预设模板记录的程序指令转换方式以及所读取的参数,将所述输入信息转换为描述所述优化模型的程序指令,包括:

基于所述转换关系,将所述运算符转换为对应的运算公式。

7. 根据权利要求1所述的方法,所述获取用户的输入信息,包括:

获取用户在用户界面中输入的输入信息;所述用户界面包括数据源输入组件和信息输入组件。

8. 根据权利要求2所述的方法,所述数据源参数基于表格型数据结构存储;所述存储

位置信息包括参数在表格中的位置信息。

9. 一种电子设备,所述电子设备包括:

处理器;

用于存储处理器可执行指令的存储器;

其中,所述处理器调用所述可执行指令时实现权利要求1-8任一所述方法的操作。

10. 一种计算机程序产品,包括计算机程序,所述计算机程序被处理器执行时实现如权利要求1-8任一所述方法的步骤。

11. 一种计算机可读存储介质,所述计算机可读存储介质上存储有若干计算机指令,所述计算机指令被执行时执行权利要求1-8任一所述的方法。

优化模型的求解方法、电子设备、应用程序及存储介质

技术领域

[0001] 本说明书实施例涉及数据处理技术领域,尤其涉及优化模型的求解方法、电子设备、应用程序及存储介质。

背景技术

[0002] 优化模型(OptimizationModel)是一种运用线性规划、非线性规划、动态规划、整数规划以及系统科学方法所确定的表示最优方案的模型。优化模型能反映条件极值问题,即在既定目标下,如何最有效地利用各种资源,或者在资源有限制的条件下,如何获取到最好的效果。优化模型方法常常可以用来解决资源最佳分配问题、最优部门结构问题、生产力合理布局问题、最优积累率问题、物资合理调运问题、最低成本问题等实际业务问题。

[0003] 在利用优化模型方法解决实际业务问题时,包括优化模型建立与优化模型求解两个过程。其中,在优化模型的建立过程中,技术人员首先要推导出优化模型,然后再使用建模语言(如PuLP语言和Pyomo语言)将优化模型编写成计算机程序,以使计算机对优化模型进行求解。显然,优化模型的建立要求技术人员掌握建模语言,模型建立门槛较高。同时技术人员需要编写优化模型对应的计算机程序,需要花费较长时间,降低了业务问题解决效率。

发明内容

[0004] 本说明书实施例提供了优化模型的求解方法、电子设备、应用程序及存储介质,可以提高业务问题解决效率。

[0005] 根据本说明书实施例的第一方面,提供一种优化模型的求解方法,所述优化模型包括决策变量、优化目标和约束条件;所述方法包括:

获取描述所述优化模型的程序指令;所述程序指令包括:用于描述所述决策变量的程序指令、用于描述所述优化目标的程序指令、以及用于描述所述约束条件的程序指令;所述程序指令携带参数;

执行所述程序指令,求解出所述决策变量的值;

其中,所述程序指令的获取步骤包括:

获取用户的输入信息,所述输入信息包括:用于描述所述决策变量的信息、用于描述所述优化目标的信息、以及用于描述所述约束条件的信息;所述输入信息携带所述参数的存储位置信息;

基于所述存储位置信息,读取所述参数;

根据预设模板记录的程序指令转换方式以及所读取的参数,将所述输入信息转换为描述所述优化模型的程序指令。

[0006] 在一些例子中,所述参数存储在数据源中,所述输入信息携带所述参数的存储位置信息;包括:

所述输入信息携带数据源标识和所述存储位置信息;所述数据源标识用于表示参

数从数据源中确定。

[0007] 在一些例子中,所述用于描述所述优化目标的信息,包括优化目标表达式;所述优化目标表达式包括预定义的优化目标字符串;

所述用于描述所述约束条件的信息,包括用于约束条件表达式;所述约束条件表达式包括预定义的约束条件字符串。

[0008] 在一些例子中,所述用于描述所述决策变量的信息,包括:

决策变量表达式;所述决策变量表达式包括预定义的决策变量字符串;和/或赋值所述决策变量的参数的存储位置信息。

[0009] 在一些例子中,在所述用于描述所述决策变量的信息包括赋值所述决策变量的参数的存储位置信息的情况下,所述执行所述程序指令,求解出所述决策变量的值,包括:

在执行所述程序指令时,基于赋值所述决策变量的参数的存储位置,读取赋值所述决策变量的参数;

基于读取的参数,求解所述决策变量的值。

[0010] 在一些例子中,所述输入信息携带运算符;所述预设模板还记录有所述运算符与运算公式的转换关系,所述根据预设模板记录的程序指令转换方式以及所读取的参数,将所述输入信息转换为描述所述优化模型的程序指令,包括:

基于所述转换关系,将所述运算符转换为对应的运算公式。

[0011] 在一些例子中,所述获取用户的输入信息,包括:

获取用户在用户界面中输入的输入信息;所述用户界面包括数据源输入组件和信息输入组件。

[0012] 在一些例子中,所述数据源参数基于表格型数据结构存储;所述存储位置信息包括参数在表格中的位置信息。

[0013] 根据本说明书实施例的第二方面,提供一种电子设备,所述电子设备包括:

处理器;

用于存储处理器可执行指令的存储器;

其中,所述处理器调用所述可执行指令时实现上述第一方面任一所述方法的操作。

[0014] 根据本说明书实施例的第三方面,提供一种计算机程序产品,包括计算机程序,所述计算机程序被处理器执行时实现上述第一方面任一所述方法的步骤。

[0015] 根据本说明书实施例的第四方面,提供一种计算机可读存储介质,所述计算机可读存储介质上存储有若干计算机指令,所述计算机指令被执行时执行上述第一方面任一所述的方法。

[0016] 本说明书实施例的实施例提供的技术方案可以包括以下有益效果:

本说明书实施例提供了优化模型的求解方法、电子设备、应用程序及存储介质,优化模型的求解过程包括获取描述优化模型的程序指令,其中,程序指令中携带有参数。然后执行该程序指令以求解出优化模型中决策变量的值。在获取程序指令的过程中,用户可以输入一些输入信息,包括描述决策变量的信息,描述优化目标的信息以及描述约束条件的信息。其中,输入信息携带有参数的存储位置信息。如此,可以基于存储位置信息读取到程序指令中的参数,并且根据预设模板记录的程序指令转换方式以及所读取的参数,将输入

信息转换为描述优化模型的程序指令。由于用户只需输入上述用于描述决策变量、优化目标、约束条件的信息,不需要使用建模语言将优化模型编写成计算机程序,因此用户可以不用掌握建模语言,降低了模型建立门槛。同时节省了用户编写计算机程序的时间,因此提高了业务问题解决效率。

[0017] 应当理解的是,以上的一般描述和后文的细节描述仅是示例性和解释性的,并不能限制本说明书实施例。

附图说明

[0018] 此处的附图被并入说明书中并构成本说明书实施例的一部分,示出了符合本说明书实施例的实施例,并与说明书一起用于解释本说明书实施例的原理。

[0019] 图1是相关技术中优化模型建立与求解过程的示意图。

[0020] 图2是本说明书根据一实施例示出的优化模型的求解方法的流程图。

[0021] 图3是本说明书根据另一实施例示出的优化模型的求解方法的流程图。

[0022] 图4A是本说明书根据一实施例示出的用户界面的示意图。

[0023] 图4B是本说明书根据另一实施例示出的用户界面的示意图。

[0024] 图5是本说明书根据一实施例示出的程序指令的示意图。

[0025] 图6是本说明书根据一实施例示出的一种电子设备的硬件结构图。

具体实施方式

[0026] 这里将详细地对示例性实施例进行说明,其示例表示在附图中。下面的描述涉及附图时,除非另有表示,不同附图中的相同数字表示相同或相似的要素。以下示例性实施例中所描述的实施方式并不代表与本说明书实施例相一致的所有实施方式。相反,它们仅是与如所附权利要求书中所详述的、本说明书实施例的一些方面相一致的装置和方法的例子。

[0027] 在本说明书实施例使用的术语是仅仅出于描述特定实施例的目的,而非旨在限制本说明书实施例。在本说明书实施例和所附权利要求书中所使用的单数形式的“一种”、“所述”和“该”也旨在包括多数形式,除非上下文清楚地表示其他含义。还应当理解,本文中使用的术语“和/或”是指并包含一个或多个相关联的列出项目的任何或所有可能组合。

[0028] 应当理解,尽管在本说明书实施例可能采用术语第一、第二、第三等来描述各种信息,但这些信息不应限于这些术语。这些术语仅用来将同一类型的信息彼此区分开。例如,在不脱离本说明书实施例范围的情况下,第一信息也可以被称为第二信息,类似地,第二信息也可以被称为第一信息。取决于语境,如在此所使用的词语“如果”可以被解释成为“在……时”或“当……时”或“响应于确定”。

[0029] 优化模型(OptimizationModel)是一种运用线性规划、非线性规划、动态规划、整数规划以及系统科学方法所确定的表示最优方案的模型。优化模型能反映条件极值问题,即在既定目标下,如何最有效地利用各种资源,或者在资源有限制的条件下,如何获取到最好的效果。优化模型方法常常可以用来解决资源最佳分配问题、最优部门结构问题、生产力合理布局问题、最优积累率问题、物资合理调运问题、最低成本问题等实际业务问题。

[0030] 优化模型一般包括决策变量、优化目标和约束条件。其中,决策变量是指实际业务

问题中涉及的与约束条件和优化目标有关的待求解的变量。优化目标是指与决策变量有关的待求其极值的函数,用于指导决策变量的求解过程。约束条件是指在求优化目标的极值时,决策变量必须满足的限制。

[0031] 作为例子,在一个资源分配问题中,某公司可以生产产品A与产品B,其中,产品A的售价为 P_1 ,原料成本为 C_1 ,所需生产工时为 n_1 ,所需修整工时为 m_1 ;产品B的售价为 P_2 ,原料成本为 C_2 ,所需生产工时为 n_2 ,所需修整工时为 m_2 。可用的生产工时为 N ,可用的修整工时为 M ,产品A的生产数量不超过 Q 。求问如何设定产品A与产品B的生产数量可以最大化该公司的收益。

[0032] 为了解决上述资源分配问题,可以建立优化模型。其中,决策变量可以包括产品A的生产数量 X_1 ,与产品B的生产数量 X_2 。优化目标包括最大化公司的收益 Z ,即 $\max Z = (P_1 - C_1) * X_1 + (P_2 - C_2) * X_2$ 。约束条件包括: $n_1 * X_1 + n_2 * X_2 \leq N$; $m_1 * X_1 + m_2 * X_2 \leq M$; $X_1 \leq Q$ 。通过求解上述优化模型,得到决策变量的取值,可以解决上述资源分配问题。

[0033] 如图1所示,在利用优化模型方法解决实际业务问题时,技术人员首先从数据源中要读取建立优化模型所需的数据,例如决策变量、优化目标和约束条件中的参数。如在上述例子中,需要在数据源中读取优化目标中的参数 P_1 、 P_2 、 C_1 、 C_2 ,以及约束条件中的参数 n_1 、 n_2 、 N 、 m_1 、 m_2 、 M 、 Q 。在推导出优化模型后再使用建模语言(如PuLP语言和Pyomo语言等)将优化模型编写成计算机程序。最后计算机调用求解器读取计算程序并求解出决策变量,得到优化模型的解。显然,优化模型的建立要求技术人员掌握建模语言,模型建立门槛较高。同时技术人员需要自己去读取查询数据,手动编写优化模型对应的计算机程序,需要花费较长时间,降低了业务问题解决效率。

[0034] 为此,本说明书实施例提出了一种优化模型的求解方法,用于建立与求解优化模型,其中,优化模型包括决策变量、优化目标和约束条件。上述方法包括如图2-图3所示的步骤:

步骤210:获取描述所述优化模型的程序指令;所述程序指令携带参数;

其中,所述程序指令包括:用于描述所述决策变量的程序指令、用于描述所述优化目标的程序指令、以及用于描述所述约束条件的程序指令。

[0035] 具体地,程序指令的获取步骤包括:

步骤211:获取用户的输入信息,所述输入信息携带所述参数的存储位置信息;

其中,所述输入信息包括:用于描述所述决策变量的信息、用于描述所述优化目标的信息、以及用于描述所述约束条件的信息;

步骤212:基于所述存储位置信息,读取所述参数;

步骤213:根据预设模板记录的程序指令转换方式以及所读取的参数,将所述输入信息转换为描述所述优化模型的程序指令。

[0036] 在获取描述优化模型的程序指令后,执行以下步骤:

步骤220:执行所述程序指令,求解出所述决策变量的值。

[0037] 其中,用户可以是上述的技术人员等进行优化模型建立与求解的任何人员。

[0038] 如上所述,优化模型包括决策变量、优化目标和约束条件这三要素。在求解优化模型的过程中,需要使用程序指令来描述上述三要素,以使计算机通过执行程序指令来求解出决策变量的值。

[0039] 其中,约束条件可以包括多个,针对每个约束条件,都可以有相应的程序指令。如在上述资源分配问题中,建立的优化模型共包括3个约束条件,则用于描述该优化模型的约束条件的程序指令可以共有3条。

[0040] 程序指令携带有参数,例如用于描述优化目标的程序指令和用于描述约束条件的程序指令都携带有参数。如在上述例子中,优化目标 $\max Z=(P_1-C_1)*X_1+(P_2-C_2)*X_2$ 包括有参数 P_1 、 C_1 与 P_2 、 C_2 ,因此,用于描述优化目标的程序指令同样携带有参数 P_1 、 C_1 与 P_2 、 C_2 。约束条件 $n_1*X_1+n_2*X_2\leq N$ 包括有参数 n_1 与 n_2 ;约束条件 $m_1*X_1+m_2*X_2\leq M$ 包括有参数 m_1 与 m_2 。因此,用于描述约束条件的程序指令同样携带有相应的参数。

[0041] 程序指令可以通过用户的输入信息转换得到,其中,输入信息携带有上述参数的存储位置信息。在一些实施例中,参数可以存储在数据源中,例如可以存储在远程存储的数据源中。远程存储的数据源可以包括但不限于ODPS(OpenDataProcessingService,开放数据处理服务)数据库,MySQL(关系型数据库管理系统)数据库等。如此,参数的存储位置信息可以包括参数在数据源中的存储位置信息。在一些实施例中,参数可以按照一定的数据格式存储在数据源中,例如数组、表格等。作为例子,参数可以基于表格型数据结构存储在数据源中。那么,参数的存储位置信息可以包括参数所在表格在数据源中的存储位置信息,以及参数在表格中的位置信息。

[0042] 预设模板记录有程序指令的转换方式,例如,预设模板可以记录有将用于描述决策变量的信息转换为用于描述决策变量的程序指令的方式、将用于描述优化目标的信息转换为用于描述优化目标的程序指令的方式、以及将用于描述约束条件的信息转换为用于描述约束条件的程序指令的方式。由于用户的输入信息携带有参数的存储位置信息,因此可以基于存储位置信息读取到程序指令携带的参数。同时预设模板中还记录有上述三种程序指令的转换方式,因此通过转换方式以及所读取的参数,可以将输入信息转换为程序指令。

[0043] 与相关技术中需要用户手动编写程序指令的过程相比,本说明书实施例提供的方法中只需用户输入描述上述三要素的输入信息即可。计算机可以自动从输入信息中读取程序指令所携带的参数,以及将输入信息转换成相应的程序指令。因此,用户不需要掌握建模语言,降低了模型建立门槛。同时节省了手动编写计算机程序的时间,因此提高了业务问题解决效率。

[0044] 如上所述,参数可以存储在数据源中,输入信息可以携带有参数的存储位置信息,那么为了让计算机能识别到输入信息中的存储位置信息,在一些实施例中,输入信息可以携带有数据源标识和上述存储位置信息。其中,数据源标识用于表示参数从数据源中确定,存储位置信息用于查询数据源中的参数。数据源标识可以是预先定义好的标识,当计算机识别到数据源标识时,表示跟随在数据源标识后的字符串为参数的存储位置信息。在一些实施例中,数据源标识可以包括起始标识和参数输入标识。起始标识用于定义决策变量和/或参数,是一个预定的符号标识。参数输入标识用于表示参数的输入操作。作为例子,起始标识可以包括但不限于符号“@”、符号“#”、符号“%”中的一种或多种的组合。参数输入标识可以包括但不限于“input”。当然,起始标识与参数标识并不限于上述举例,还可以是形式的标识,本申请在此不做限制。

[0045] 例如,输入信息可以包括参数所在表格在数据源中的存储位置信息,以及包括“@input.type”,其中,“@input”为由起始标识“@”和参数输入标识“input”组成的数据源标

识。那么,跟随在数据源标识“@input”后的字符串“type”为参数在表格中的位置信息。如此,基于参数所在表格在数据源中的存储位置信息,以及参数在表格中的位置信息,可以获取到参数。

[0046] 输入信息可以包括用于描述决策变量的信息、用于描述优化目标的信息、以及用于描述约束条件的信息。其中,用于描述优化目标的信息可以包括优化目标表达式。所谓表达式,是指由数字、算符、数字分组符号(如括号)、自由变量、和约束变量等以能求得数值的有意义排列方式所得的组合。其中,约束变量在表达式中已被指定数值,自由变量则未指定数值。所谓优化目标表达式,即用于描述优化目标的表达式。其中,为了让计算机识别到用户输入的优化目标表达式是用于描述优化目标,优化目标表达式可以包括有预定义的优化目标字符串。当计算机识别到优化目标字符串时,可以确定该字符串所在的表达式是用于描述优化目标的。作为例子,预定义的优化目标字符串可以是字符串“obj”。当然,预定义的优化目标字符串不限于上述举例,还可以是其他字符串,本申请在此不做限制。

[0047] 同样地,用于描述约束条件的信息可以包括约束条件表达式。所谓约束条件表达式,即用于描述约束条件的表达式。如上所述,约束条件可以包括有多个,针对每个约束条件,用户都可以输入相应的表达式。其中,为了让计算机识别到用户输入的约束条件表达式是用于描述约束条件,约束条件表达式可以包括有预定义的约束条件字符串。当计算机识别到约束条件字符串时,可以确定该字符串所在的表达式是用于描述约束条件的。作为例子,预定义的约束条件字符串可以是字符串“c_n”,其中,n可以是正整数。当然,预定义的约束条件字符串不限于上述举例,还可以是其他字符串,本申请在此不做限制。

[0048] 用于描述决策变量的信息可以包括决策变量表达式。也即,用户可以直接输入表达式来描述决策变量。其中,为了让计算机识别到用户输入的决策变量表达式是用于描述决策变量,决策变量表达式可以包括有预定义的决策变量字符串。当计算机识别到决策变量字符串时,可以确定该字符串所在的表达式是用于描述决策变量的。在一些实施例中,决策变量字符串可以包括起始标识和决策变量标识。作为例子,起始标识可以是包括但不限于符号“@”、符号“#”、符号“%”中的一种或多种的组合。决策变量标识可以是X。当然,决策变量标识可以不限于上述举例,本申请在此不做限制。决策变量字符串可以是字符串“@X”。

[0049] 在上述实施例中,用户输入的信息为分别用于描述决策变量、优化目标以及约束条件的表达式。从用户的角度上来说,相比于程序指令,表达式有更强的可读性,也更便于用户编写。但从计算机的角度上来说,计算机在读取用户输入的多条表达式时,无法分辨到每条表达式所描述的内容。对此,本实施例提出了可以预定义不同的字符串,包括决策变量字符串、优化目标字符串以及约束条件字符串。当计算机读取到这些预定义的字符串时,便能识别出该预定义字符串所在表达式所描述的内容。

[0050] 此外,用于描述决策变量的信息还可以是赋值给决策变量的参数的存储位置信息。作为例子,参数可以按照表格的格式存储在数据源中,那么输入信息可以包括参数所在表格在数据源中的存储位置信息,用于描述决策变量的信息可以包括参数在表格中的位置信息。例如,字符串“type”为赋值给决策变量的参数在表格中的位置信息,那么用于描述决策变量的信息可以包括字符串“type”。

[0051] 在一些实施例中,当用于描述决策变量的信息包括赋值给决策变量的参数的存储位置时,那么在步骤220中决策变量的求解过程可以包括:在执行程序指令时,基于赋值决

策变量的参数的存储位置,读取该参数;并基于所读取的参数,求解决策变量的值。如在上述例子中,用于描述决策变量的信息包括字符串“type”,表示表格中字段为type的参数用于赋值给决策变量。如此,在求解决策变量的过程中,可以从表格中读取字段为type的参数作为决策变量,并进行求解。

[0052] 在一些实施例中,用户输入的输入信息可以携带运算符号。运算符号可以包括如但不限于求和符号SUM、均值符号AVERAGE等。上述预设模板中还记录有运算符号与运算公式的转换关系。如此,可以基于该转换关系,将运算符号转换为对应的运算公式。

[0053] 在一些实施例中,用户可以在用户界面中输入上述输入信息。图4A-图4B示出了两种用户界面的示意图。如图4A所示,用户界面400可以包括数据源输入组件410和信息输入组件420。其中,这两种输入组件可以是下拉菜单(如图4B中的决策变量组件421),也可以是输入框(如图4A中的信息输入组件420)。用户可以在下拉菜单中选择上述输入信息,也可以直接在输入框中直接输入上述输入信息。以上述资源分配问题的建模为例,参数可以按照如表1所示的表格格式存储在数据源中。数据源输入组件410可以用于从数据源中选择参数所在的表格1,例如可以按照表格1在数据源中的存储位置进行选择,也可以按照表格1的名字进行选择。作为例子,用户可以在数据源输入组件410的下拉菜单中查找表格1在数据源中的存储位置,以选定表格1;也可以直接在数据源输入组件410的输入框中输入表格1的名字、唯一标识符等其他能唯一表征表格1的标识。

[0054] 表1

type	price	material cost	finishing	carpentry	max num
产品 A	P ₁	C ₁	n ₁	m ₁	Q
产品 B	P ₂	C ₂	n ₂	m ₂	/

如图4A所示,信息输入组件420可以用于输入优化模型的决策变量表达式、优化目标表达式以及约束条件表达式。如在上述资源分配问题中,决策变量表达式可以包括@X=@input.type。其中,@为起始标识,X为决策变量标识,@X为决策变量字符串。当计算机识别到决策变量字符串@X时,可以确定该表达式是用于描述决策变量的。input为参数输入标识,@input为数据源标识,type为赋值决策变量的参数在表格中的位置信息。其中,位置信息包括表格中的字段。结合表1可知,该决策变量表达式表示将表1中字段为type的参数(产品A与产品B)作为优化模型的决策变量。当计算机识别到数据源标识@input时,可以根据字段type在表1中确定赋值给决策变量的参数。

[0055] 在上述资源分配问题中,优化目标表达式可以包括obj = sum([(@input.price [i] - @input.materials_cost[i]) * @X[i] for i in @X])。其中,obj为优化目标字符串,当计算机识别到优化目标字符串obj时,可以确定该表达式是用于描述优化目标的。sum为表示求和的运算符号,根据预设模板记录的运算符号与运算公式的转换关系,可以将该运算符号转换成对应的运算公式。该优化目标表达式表示每种产品的利润之和。某一种产品的利润为(售价-成本)*生产数量。当计算机识别到数据源标识@input时,可以根据字段price以及materials_cost从表1中确定相应的参数的取值。

[0056] 在上述资源分配问题中,约束条件表达式可以包括3条,第一条约束条件表达式可以包括c1 = sum([(@input.finishing[i] * @X[i] for i in @X]) <= N。第二条约束条

件表达式可以包括 $c2 = \text{sum}([\text{@input.carpentry}[i] * \text{@X}[i] \text{ for } i \text{ in } \text{@X}]) \leq M$ 。第三条约束条件表达式可以包括 $c3 = \text{for } i \text{ in } \text{@X} : \text{@X}[i] \leq \text{@input.max_num}[i]$ 。其中,c1、c2、和c3为约束条件字符串,当计算机识别到约束条件字符串时,可以确定该表达式是用于描述约束条件的。第一条约束条件表达式表示每种产品所需生产工时之和不大于N。第二条约束条件表达式表示每种产品所需修正工时之和不大于M。第三约束条件表达式表示每种产品的生产数量不大于对应的最大值。当计算机识别到数据源标识@input时,可以根据字段finishing、carpentry以及max_num从表1中确定相应的参数的取值。

[0057] 在一些实施例中,如图4B所示,信息输入组件420可以包括决策变量输入组件421、优化目标输入组件422以及约束条件输入组件423。决策变量输入组件421可以用于输入赋值给决策变量的参数的存储位置信息以及用于定义决策变量的类型。其中,决策变量的类型可以包括但不限于:整数(Integers)、正整数(Positive Integers)、非正整数(Non-Positive Integers)、负整数(Negative Integers)、非负整数(Non-Negative Integers)、实数(Reals)、正实数(Positive Reals)、非正实数(Non-Positive Reals)、负实数(Negative Reals)、非负实数(Non-Negative Reals)等。如在上述资源分配问题中,可以在决策变量输入组件421中输入字段“type”以及在下拉菜单下选择决策变量的类型为非负整数。

[0058] 优化目标输入组件422可以用于输入用于描述优化目标的信息。如在上述资源分配问题中,用户可以直接在优化目标输入组件中输入 $\text{sum}([\text{@input.price}[i] - \text{@input.materials_cost}[i]) * \text{@X}[i] \text{ for } i \text{ in } \text{@X}])$ 。由于计算机可以直接识别优化目标输入组件422中输入的信息是用于描述优化目标的,因此用户可以无需输入携带优化目标字符串的表达式。

[0059] 约束条件输入组件423可以用于输入描述约束条件的信息。如在上述资源分配问题中,用户可以直接在优化目标输入组件中输入“ $\text{sum}([\text{@input.finishing}[i] * \text{@X}[i] \text{ for } i \text{ in } \text{@X}]) \leq N$ ”、“ $\text{sum}([\text{@input.carpentry}[i] * \text{@X}[i] \text{ for } i \text{ in } \text{@X}]) \leq M$ ”、以及“ $\text{for } i \text{ in } \text{@X} : \text{@X}[i] \leq \text{@input.max_num}[i]$ ”。由于计算机可以直接识别约束条件输入组件423中输入的信息是用于描述约束条件的,因此用户可以无需输入携带约束条件字符串的表达式。

[0060] 在获取到用户输入的输入信息后,通过上述任一实施例记载的方法可以读取到程序指令中的参数。另外,预设模板中还记录有程序指令的转换方式,可以将输入信息转换为程序指令。如在上述资源分配问题中,通过预设模板可以将输入信息转换为如图5所示的程序指令。

[0061] 本说明书实施例提供了优化模型的求解方法,优化模型的求解过程包括获取描述优化模型的程序指令,其中,程序指令中携带有参数。然后执行该程序指令以求解出优化模型中决策变量的值。在获取程序指令的过程中,用户可以输入一些输入信息,包括描述决策变量的信息,描述优化目标的信息以及描述约束条件的信息。其中,输入信息携带有参数的存储位置信息。如此,可以基于存储位置信息读取到程序指令中的参数,并且根据预设模板记录的程序指令转换方式以及所读取的参数,将输入信息转换为描述优化模型的程序指令。由于用户只需输入上述用于描述决策变量、优化目标、约束条件的信息,不需要使用建模语言将优化模型编写成计算机程序,因此用户可以不用掌握建模语言,降低了模型建立

门槛。同时节省了用户编写计算机程序的时间,因此提高了业务问题解决效率。

[0062] 基于上述任意实施例所述的优化模型的求解方法,本说明书实施例还提供了如图6所示的一种电子设备的结构示意图。如图6,在硬件层面,该电子设备包括处理器、内部总线、网络接口、内存以及非易失性存储器,当然还可能包括其他业务所需要的硬件。处理器从非易失性存储器中读取对应的计算机程序到内存中然后运行,以实现上述任意实施例所述的优化模型的求解方法。

[0063] 基于上述任意实施例所述的优化模型的求解方法,本说明书实施例还提供了一种计算机程序产品,包括计算机程序,计算机程序被处理器执行时可用于执行上述任意实施例所述的优化模型的求解方法。

[0064] 基于上述任意实施例所述的优化模型的求解方法,本说明书实施例还提供了一种计算机存储介质,存储介质存储有计算机程序,计算机程序被处理器执行时可用于执行上述任意实施例所述的优化模型的求解方法。

[0065] 上述对本说明书实施例特定实施例进行了描述。其它实施例在所附权利要求书的范围内。在一些情况下,在权利要求书中记载的动作或步骤可以按照不同于实施例中的顺序来执行并且仍然可以实现期望的结果。另外,在附图中描绘的过程不一定要求示出的特定顺序或者连续顺序才能实现期望的结果。在某些实施方式中,多任务处理和并行处理也是可以的或者可能是有利的。

[0066] 本领域技术人员在考虑说明书及实践这里申请的发明后,将容易想到本说明书实施例的其它实施方案。本说明书实施例旨在涵盖本说明书实施例的任何变型、用途或者适应性变化,这些变型、用途或者适应性变化遵循本说明书实施例的一般性原理并包括本说明书实施例未申请的本技术领域中的公知常识或惯用技术手段。说明书和实施例仅被视为示例性的,本说明书实施例的真正范围和精神由下面的权利要求指出。

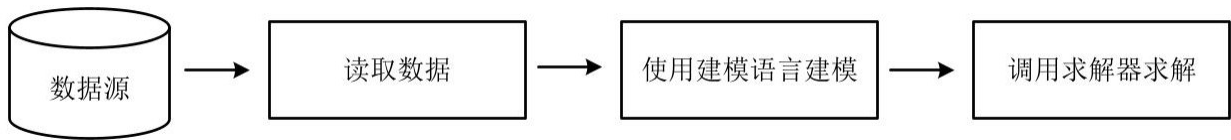


图1

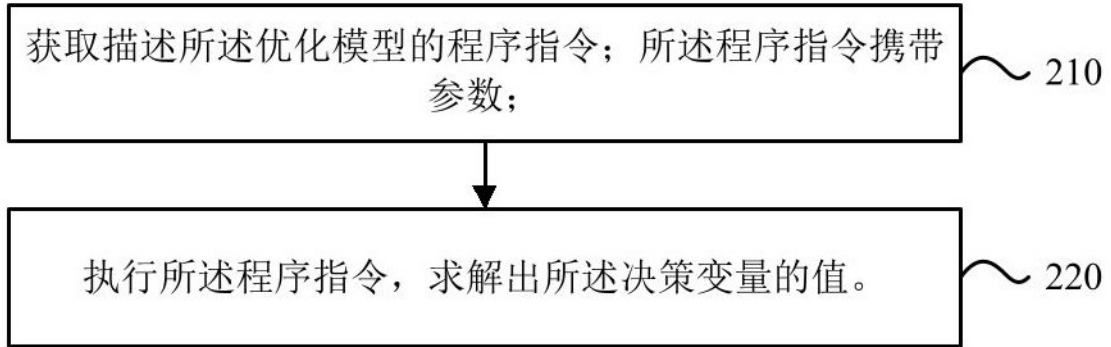


图2

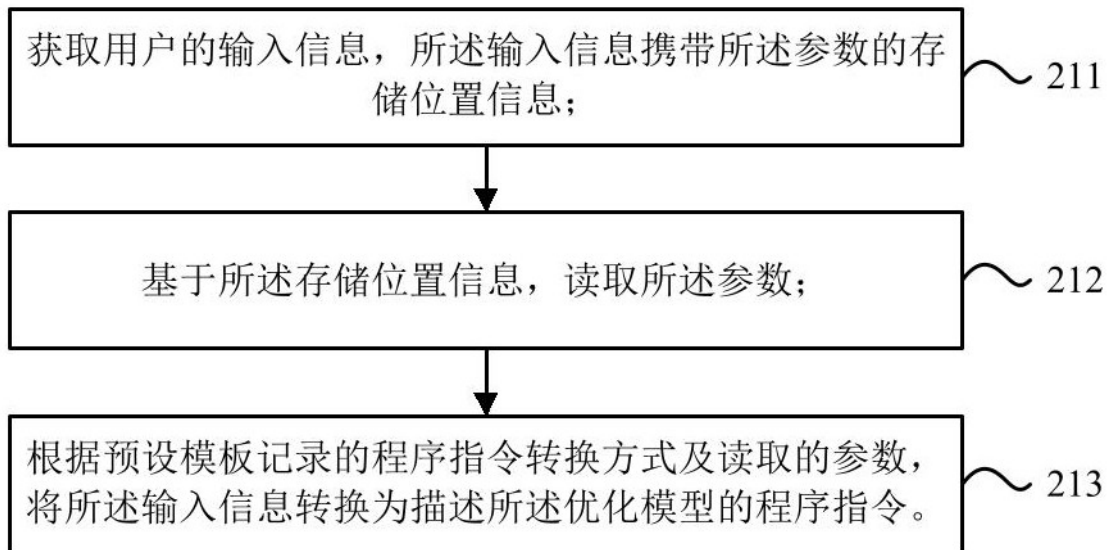


图3

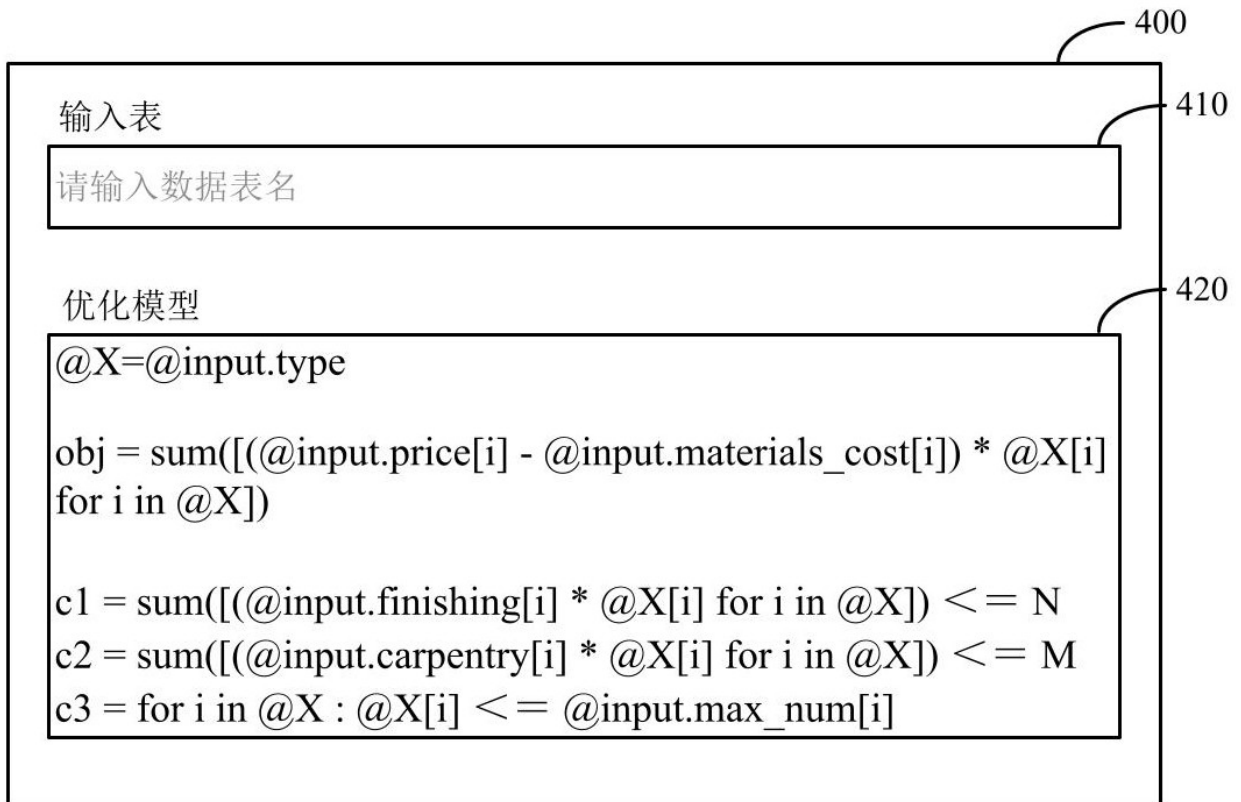


图4A

图4B

```
1  # coding: utf-8
2
3  from pyomo.opt import SolverFactory
4  from pyomo.environ import (ConcreteModel, Var, Objective, maximize,
5  Constraint, NonNegativeIntegers)
6
7
8
9
10
11 model = ConcreteModel()
12
13 # variable
14 toys = ['soldier', 'train']
15 model.x = Var(toys, within=NonNegativeIntegers)
16
17 # object
18 def objRule(model):
19     return 3 * model.x['soldier'] + 2 * model.x['train']
20 model.Obj = Objective(rule=objRule, sense=maximize)
21
22 # constraints
23 def rule1(model):
24     return 2 * model.x['soldier'] + model.x['train'] <= 100
25 model.c1 = Constraint(rule=rule1)
26
27 def rule2(model):
28     return model.x['soldier'] + model.x['train'] <= 80
29 model.c2 = Constraint(rule=rule2)
30
31 def rule3(model):
32     return model.x['soldier'] <= 40
33 model.c3 = Constraint(rule=rule3)
```

图5

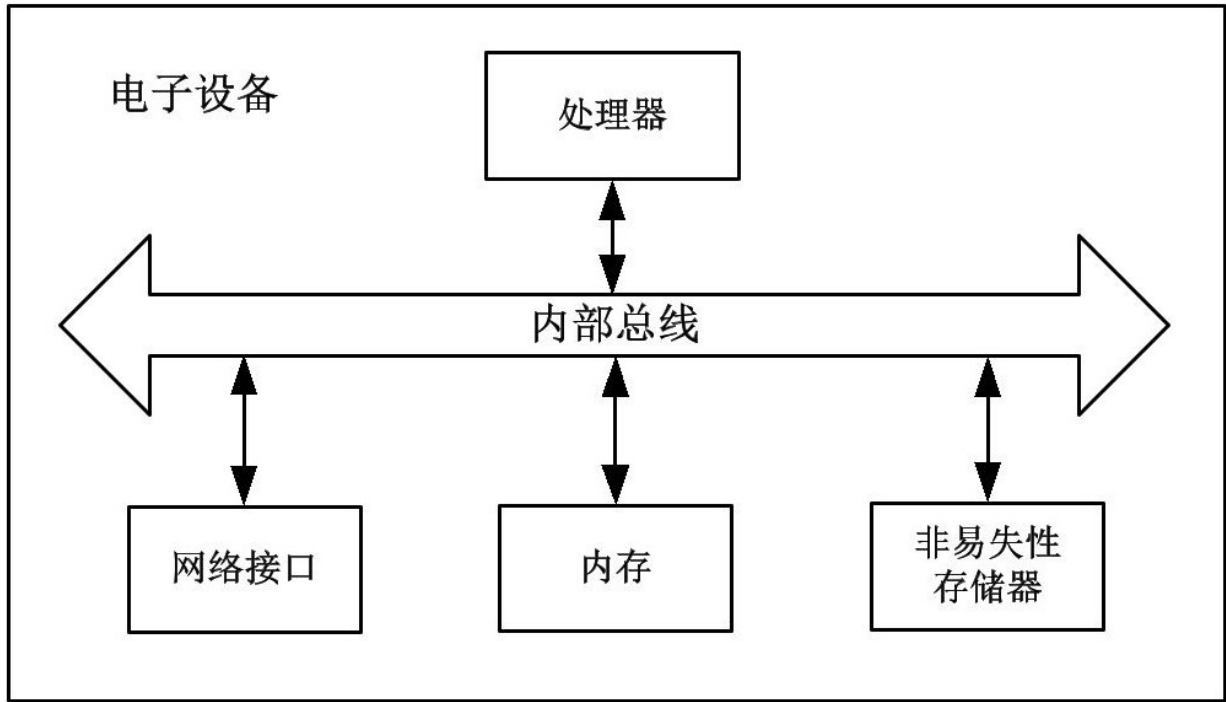


图6