**路径规划问题**

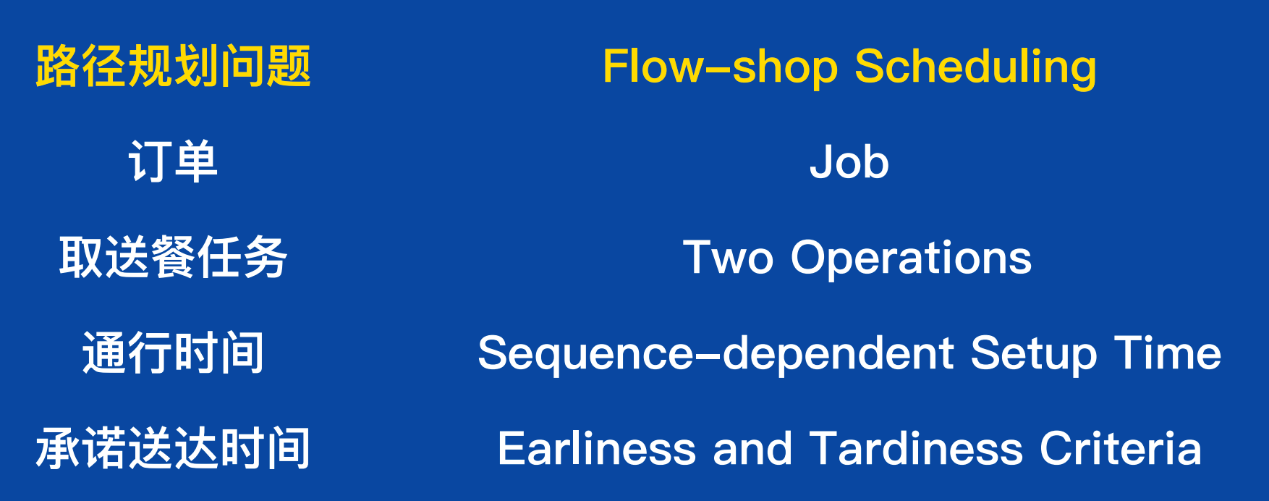
这是智能调度系统中最为重要的一个环节。系统派单、系统改派，都依赖路径规划算法。在骑手端，给每个骑手推荐任务执行顺序。另外，用户点了外卖之后，美团会实时展示骑手当前任务还需要执行几分钟，要给用户提供更多预估信息。这么多应用场景，共同的诉求是对时效要求非常高，算法运行时间要越短越好。

但是，算法仅仅是快就可以吗？并不是。因为这是派单、改派这些环节的核心模块，所以算法的优化求解能力也非常重要。如果路径规划算法不能给出较优路径，可想而知，上层的指派和改派很难做出更好的决策。

所以，对这个问题做明确的梳理，核心的诉求是优化效果必须是稳定的好。不能这次的优化结果好，下次就不好。另外，运行时间一定要短。

在求解路径规划这类问题上，很多公司的技术团队，都经历过这样的阶段：起初，采用类似遗传算法的迭代搜索算法，但是随着业务的单量变大，发现算法耗时太慢，根本不可接受。然后，改为大规模邻域搜索算法，但算法依然有很强的随机性，因为没有随机性在就没办法得到比较好的解。而这种基于随机迭代的搜索策略，带来很强的不确定性，在问题规模大的场景会出现非常多的Bad Case。另外，迭代搜索耗时太长了。主要的原因是，随机迭代算法是把组合优化问题当成一个单纯的Permutation问题去求解，很少用到问题结构特征。这些算法，求解TSP时这样操作，求解VRP时也这样操作，求解Scheduling还是这样操作，这种类似“无脑”的方式很难有出色的优化效果。

所以，在这个项目中，基本可以确定这样的技术路线。首先，只能做启发式定向搜索，不能在算法中加随机扰动。**不能允许同样的输入在不同运行时刻给出不一样的优化结果。**然后，不能用普通迭代搜索，必须把这个问题结构特性挖掘出来，做基于知识的定制化搜索。



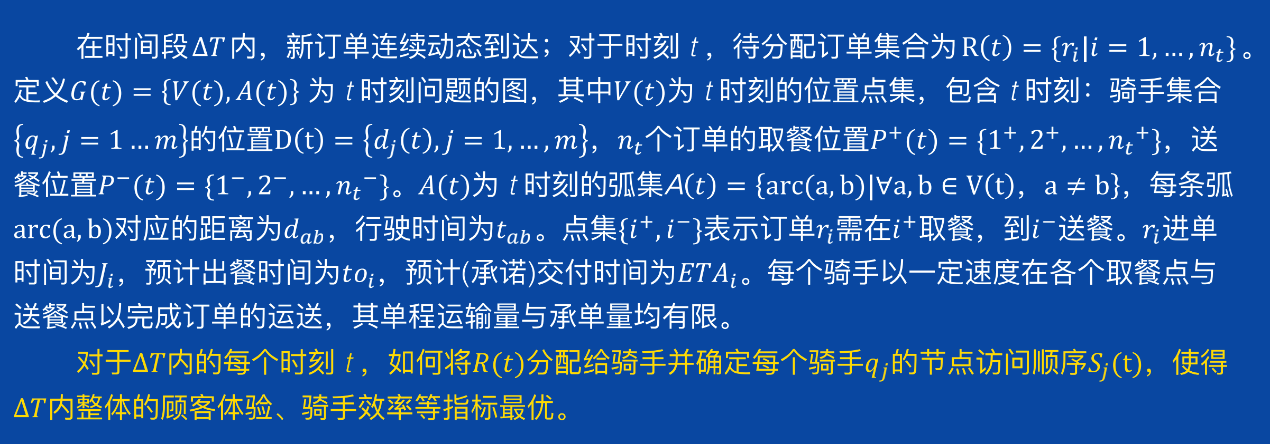
说起来容易，具体要怎么做呢？我们认为，最重要的是看待这个问题的视角。这里的路径规划问题，对应的经典问题模型，是开环TSP问题，或是开环VRP的变种么？可以是，也可以不是。我们做了一个有意思的建模转换，把它看作流水线调度问题：每个订单可以认为是job；一个订单的两个任务取餐和送餐，可以认为是一个job的operation。任意两个任务点之间的通行时间，可以认为是序列相关的准备时间。每一单承诺的送达时间，包括预订单和即时单，可以映射到流水线调度问题中的提前和拖期惩罚上。

**算法应用效果**

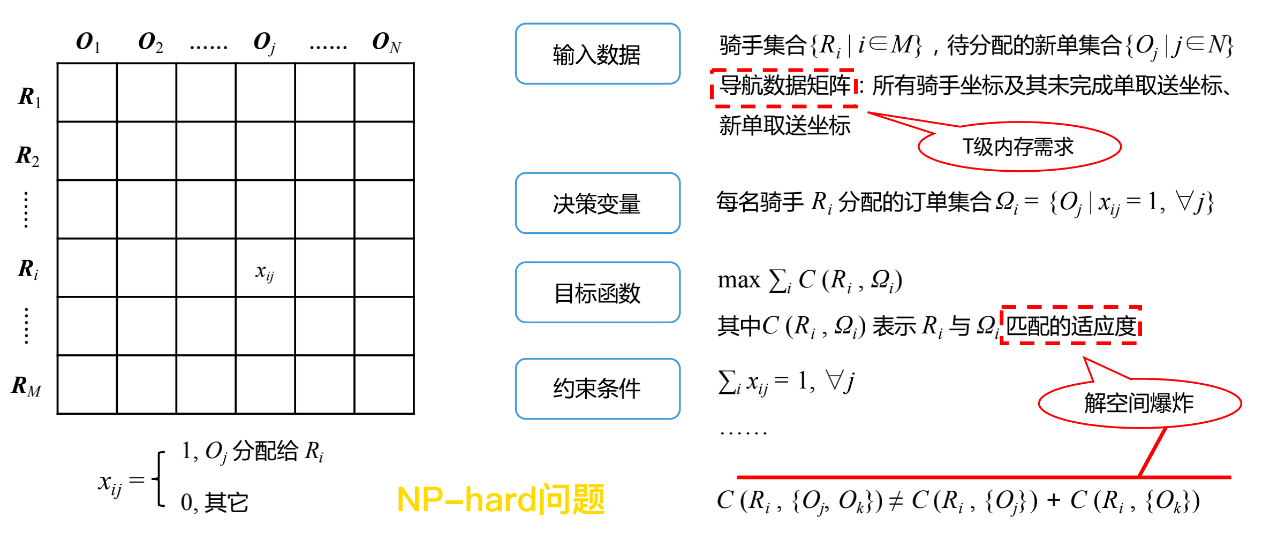
做了这样的建模转换之后，流水线调度问题就有大量的启发式算法可以借鉴。我们把一个经典的基于问题特征的启发式算法做了适当适配和改进，可以得到非常好的效果。相比于之前的算法，耗时下降70%，优化效果不错。因为这是一个确定性算法，所以运行多少次的结果都一样。我们的算法运行一次，跟其它算法运行10次的最优结果相比，优化效果是持平的。

**订单智能调度**

配送调度场景，可以用数学语言描述。它不仅是一个业务问题，更是一个标准的组合优化问题，并且是一个马尔可夫决策过程。



并非对于某个时刻的一批订单做最优分配就足够，还需要考虑整个时间窗维度，每一次指派对后面的影响。每一次订单分配，都影响了每个骑手后续时段的位置分布和行进方向。如果骑手的分布和方向不适合未来的订单结构，相当于降低了后续调度时刻最优性的天花板。所以，要考虑长周期的优化，而不是一个静态优化问题



为了便于理解，我们还是先看某个调度时刻的静态优化问题。它不仅仅是一个算法问题，还需要我们对工程架构有非常深刻的理解。因为，在对问题输入数据进行拆解的时候，会发现算法的输入数据太庞大了。比如说，我们需要任意两个任务点的导航距离数据。

而我们面临的问题规模，前几年只是区域维度的调度粒度，一个商圈一分钟峰值100多单，匹配几百个骑手，但是这种乘积关系对应的数据已经非常大了。现在，由于美团有更多业务场景，比如跑腿和全城送，是会跨非常多的商圈，甚至跨越半个城市，所以只能做城市级的全局优化匹配。目前，调度系统处理的问题的峰值规模，是1万多单和几万名骑手的匹配。而算法允许的运行时间只有几秒钟，同时对内存的消耗也非常大。

另外，配送和网约车派单场景不太一样。打车的调度是做司机和乘客的匹配，本质是个二分图匹配问题，有多项式时间的最优算法：KM算法。打车场景的难点在于，如何刻画每对匹配的权重。而配送场景还需要解决，对于没有多项式时间最优算法的情况下，如何在指数级的解空间，短时间得到优化解。如果认为每一单和每个骑手的匹配有不同的适应度，那么这个适应度并不是可线性叠加的。也就意味着多单对多人的匹配方案中，任意一种匹配都只能重新运算适应度，其计算量可想而知。



总结一下，这个问题有三类挑战：

1. 性能要求极高，要做到万单对万人的秒级求解。我们之前做了一些比较有意思的工作，比如基于历史最优指派的结果，用机器学习模型做剪枝。基于大量的历史数据，可以帮助我们节省很多无用的匹配方案评价。
2. 动态性。 作为一个MDP问题，需要考虑动态优化场景，这涉及大量的预估环节。在只有当前未完成订单的情况下，骑手如何执行、每一单的完成时刻如何预估、未来时段会进哪些结构的订单、对业务指标和效率指标产生怎样的影响……可能会觉得这是一个典型的强化学习场景，但它的难点在于决策空间太大，甚至可以认为是无限大的。目前我们的思路，是通过其它的建模转换手段进行解决。
3. 配送业务的随机因素多。比如商家的出餐时间，也许是很长时间内都无法解决的随机性。就连历史每一个已完成订单，商家出餐时间的真值都很难获得，因为人为点击的数据并不能保证准确和完整。商家出餐时刻不确定，这个随机因素永远存在，并且非常制约配送效率的提升。另外，在顾客位置交付的时间也不确定。写字楼工作日的午高峰，上电梯、下电梯的时间，很难准确进行预估。当然，我们也在不断努力让预估变得更精准，但随机性永远存在。对于骑手来说，平台没法规定每个骑手的任务执行顺序。骑手在配送过程中可以自由发挥，所以骑手执行顺序的不确定性也一直存在。为了解决这些问题，我们尝试用鲁棒优化或是随机规划的思想。但是，如果基于随机场景采样的方式，运算量又会大幅增加。所以，我们需要进行基于学习的优化，优化不是单纯的机器学习模型，也不是单纯的启发式规则，优化算法是结合真实数据和算法设计者的经验，学习和演进而得。只有这样，才能在性能要求极高的业务场景下，快速的得到鲁棒的优化方案。